The U.K. ATARI Computer Owners Club Issue 12 Price £1.00

Independent User Group

Monitor

Inside this Issue
Arrays and Matrices
Get Motorised!

ST Reviews
DB Master One
Time Bandit
Menu

8-bit Reviews
Eidolon
Boulderdash 2
Whirlinurd
Zorro
Action Biker

LLAMASOFT

presents

The **THIRD GENERATION** of Jeff Minter's unique **LIGHT SYNTHESISERS**

- MIND/MACHINE SYNERGY -

COLOURSPACE: allows the user to create **dynamic, interactive light displays** using the Atari's 320 x 200 512-colour graphics. Do it to music... your ears won't believe your eyes...

COLOURSPACE: mouse control, over 100 keyboard commands, compatible with NEOchrome images (for foreground and background displays), 20 presets, 84 definable lightforms, record mode, load and save of parformanca data from disk.

COLOURSPACE: raw 68000 power harnessed to create a completely new artform... the first true **light synthesiser**... play light like music...

COLOURSPACE. Discover why you bought your **ST.**

NOW AVAILABLE at £19.95, from retailers of ATARI software or direct from
LLAMASOFT 49 Mount Pleasant Tadley Hants
Tel: 07356 4478
SEND S.A.E. FOR CATALOGUE & NEWSLETTER

Monitor at Atari Show

At the eleventh hour the club was persuaded to take a stand at the Atari Show held at the Novotel Hammersmith in early March. And so it was that the editorial team (that's us in the credit list below) hurriedly put together a stand and made our way to the great hall of display. We set up on the day before the show opened and we brought along plenty of Monitor magazines to sell, including back issues. We also had some back issues of ANALOG, blank disks and (eventually) copies of the new Sidewinder game from Futureware. We had no idea whether the show would be a success or a dismal flop, but we were keeping our fingers crossed. Friday 10 a.m. came and the flood gates opened, people literally poured into the hall and from then on it was all hands to the pumps: we hardly had time to think as our stand was covered in people right from the start. At about 5.15 p.m. the tide turned and we had time to reflect on the day's activities. Basically we were over the moon at the response to Monitor, and it was so nice to hear many people say that they thought the magazine was very good. It was also nice to meet so many of the current membership, its good to put faces to some of the names that have been familiar to us for many years. Although we were pleased we felt that this level of visitors could not possibly last, but how wrong we were! Saturday and Sunday were just as busy, and at the end of the three days we all felt that it was well worth our being there, roll on next year!

We still have some ANALOG back issues available, if anyone is interested, we have issues 13, 14, 15 and 17, 18. They cost £1.50 each (half-price) so send a cheque made payable to the club to P.O. Box 3, Rayleigh, Essex, stating which issues you would like. In addition we have available a joystick extension cable with a 9 way socket on one end and a 9 way plug on the other. So why not sit back in your armchair and play your favourite game, you see these cables are 10 feet long! Or you could use them for I/O add-ons as they have all 9 wires connected! Price just £3.50.

CREDITS

Editor	Roy Smith
Art Editor	Peter Blackmore
Technical Editor	Ron Levy
Technical Editor	Keith Mayhew
Adventure Editor	Steve Hillen

CONTENTS

2 **Get Motorised**
Add-on circuits to run various motors

4 **8-bit Reviews**
Technicolor Dream, Eidolon, Action Biker, Boulderdash 2 and Wirlinurd.

9 **Starting from Basics**
All about Arrays and Matrices

13 **Another Boring Space Invaders Game**
Smashing little game, boring it isn't

15 **Hot Gossip**
Whispers on the Grapevine

17 **DB Master One**
Review of the database supplied with your ST

20 **ST Reviews**
Microdeal's utilities, Time Bandit and Menu+ from Metacomco

22 **Opening Out**
Part 3 explains disk file handling techniques.

25 **Software Library**
All the new programs are listed.

26 **Adventure into the Atari**
Find out how to write a simple adventure game.

31 **Cracking the Code**
Learn 6502 code with this in depth series

SUBSCRIPTION RATES

U.K. and Ireland	£4.00
Europe	£7.00
Outside Europe (Surface)	£7.00
Outside Europe (Airmail)	£10.50

The above prices are in English Pounds Sterling and include postage and packing.

A subscription membership fee to join the U.K. Atari Computer Owners Club is just £4.00 for four issues of the club magazine. All cheques, postal orders are to be made payable to the U.K. Atari Computer Owners Club. Overseas membership is also available at slightly higher rates. Overseas members who use the Library service should include enough extra money to cover return postage.

ADVERTISEMENTS

Please note that the club cannot be held legally responsible for claims made by advertisers.

Cover: Eidolon from Activision and Time Bandit from Metacomco

Club Address: The U.K. Atari Computer Owners Club, P.O. Box 3, Rayleigh, Essex.

Copyright The UK ATARI COMPUTER OWNERS CLUB is an independent users group and is in no way affiliated with ATARI. All material is subject to world wide Copyright protection and is reproduced only on information in whole or part is expressly forbidden. All reasonable care is taken to ensure accuracy in preparation of the magazine but the UK ATARI COMPUTER OWNERS CLUB cannot be held legally responsible for its contents. Where errors occur corrections will be published as soon as possible. Permission to reproduce any part of this publication must be obtained from the UK ATARI COMPUTER OWNERS CLUB. ATARI (and other Atari products mentioned in this magazine) is a trademark of ATARI CORPORATION.

GET MOTORISED!

by Mark Brighton

DC Motor

Many of you will have seen or heard about mice and turtles They whizz through mazes or scuttle around the floor almost like living creatures providing endless fascination with their antics especially for the kids

They are in fact, an excellent means of introduction to computing for young people who will then gain a good understanding of functional programming for robotic control and develop their concept of artificial intelligence by experimentation, and the Atari 400/800 computer is ideally suited to controlling this type of simple robot having 16 (8 for XL owners) programmable control lines available at the joystick sockets Some of these lines could be used to control motors, and some to read information from sensors, etc, which would give the computer information about the robot's surroundings and so enable it to make a decision regarding what to do next.

Of the different types of motors suitable for control by the Atari joystick ports, the DC motor is the simplest to interface, requiring only a simple external transistor switch to enable the computer to control the motor power supply The external power supply is necessary because the joystick sockets alone cannot supply the power required to run the motor Such a circuit is shown in Figure 1, and when plugged into Port 1, the motor may be turned on and off using Listing 1

Two transistors are used because the first transistor provides the current to switch the motor control transistor hard on from the +12V supply, as the high impedance joystick line could not supply this current alone However this circuit has one serious drawback in that it can only make the motor run in one direction, being incapable of reversing the supply to the motor Where bi-directional operation is required, the bridge circuit of Figure 2 may be used It may be seen that, depending on which pair of transistors are turned on, the polarity of the supply connected to the motor is controllable Note that the control inputs should never be high together, as this results in a short circuit of the motor power supply Listing 2 is a suitable program to drive a DC motor in both directions The preceding circuits have shown simple ways to connect a DC motor to your Atari.

Listing 1

```
HQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CA 40 POKE PORT,0
AC 50 FOR DELAY=1 TO 400
KB 60 NEXT DELAY
CR 70 POKE PORT,1
AF 80 FOR DELAY=1 TO 400
KJ 90 NEXT DELAY
QG 100 GOTO 40
```

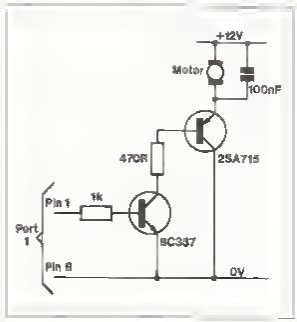


Figure 1

Servo

However, the DC motor has one major disadvantage in robotic and machine applications, which is that its rotation is only turned on or off for an arbitary period with no control over the actual position of the rotor, either during rotation or afterwards This is because a DC motor cannot be controlled down to speeds where *part* of a rotation alone can occur since it has no means of stopping the rotor in a particular place, or moving it through a precise angle

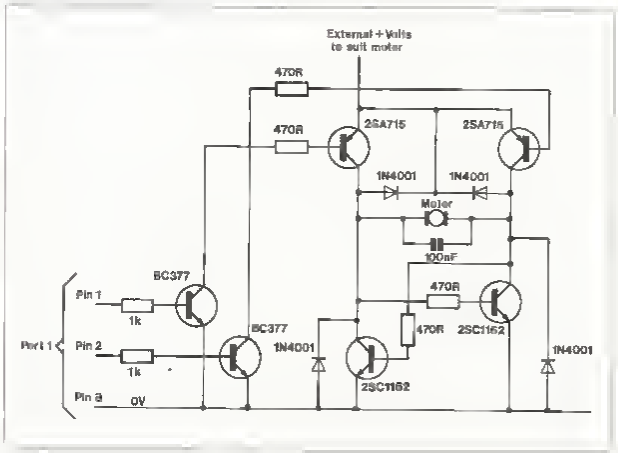


Figure 2.

The traditional answer to this problem has been to use a gearbox to reduce the final drive speed, and to increase both the holding and driving torque combined with some form of positional sensor such as a potentiometer This is the standard set-up for the type of servos used in radio-controlled models, and may be used in a computer controlled set up, assuming that some means is available to turn the analogue voltage information returned by the potentiometer into a digital format, which your computer can understand the Atari 400/800 computer is, of course, equipped with a grand total of eight analogue to digital converters (XL owners read as 4), which are ideally suited to this task An example wiring diagram is shown in Figure 3, and a test program to put a servo through its paces is included, see Listing 3 As before, an external power supply will be necessary to run the servo motor A suitable circuit is shown in Figure 4

Stepper Motor

Finally, a motor which is eminently suitable for digital control and small scale robotic/machine designs is the stepping motor The motor has several separate stator coils which are arranged in such a manner that various combinations of energised coils cause the rotor to 'step' around by a small and

Listing 2.

```
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
BS 30 POKE 54018,60
CA 40 POKE PORT,0
AC 50 FOR DELAY=1 TO 400
KG 60 NEXT DELAY
CR 70 POKE PORT,1
AF 80 FOR DELAY=1 TO 400
KJ 90 NEXT DELAY
MT 100 POKE PORT,2
XR 110 FOR DELAY=1 TO 400
SP 120 NEXT DELAY
QM 130 GOTO 40
```

Listing 3

```
ZG 5 REM ** HIGH AND LOW ARE THE
JF 6 REM ** MAX AND MIN VALUES READ
RQ 7 REM ** AT THE PADDLE PORT FOR
HW 8 REM ** THE EXTREMES OF TRAVEL
KL 9 REM ** OF YOUR SERVO.
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CO 40 POKE PORT,1
KC 50 IF PADDLE(0)<HIGH THEN 50
DF 70 POKE PORT,2
LN 80 IF PADDLE(0)>LOW THEN 80
LP 100 POKE PORT,0
XR 110 FOR DELAY=1 TO 400
GP 120 NEXT DELAY
OR 130 GOTO 40
```

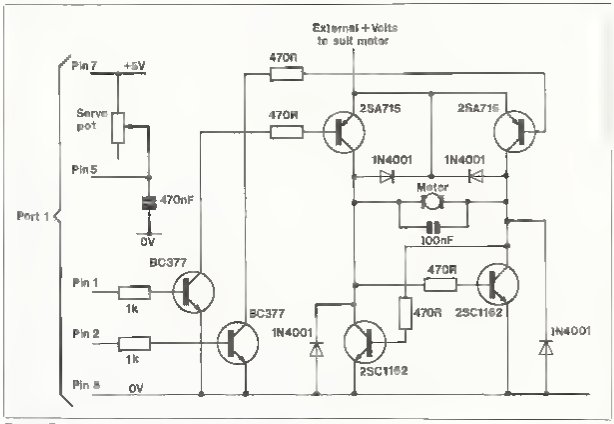


Figure 3

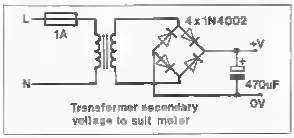


Figure 4

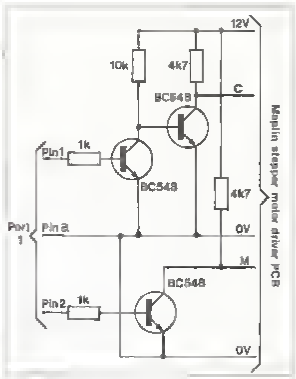


Figure 5

definite amount, e g 7 5° For this reason, the stepper motor offers great flexibility in use being completely controllable both in speed and the angle through which it turns. It also holds each new position it attains strongly, since the stator coils may remain energised even after the rotor has stopped turning It may therefore, be used directly in most applications requiring no gearbox and no means of positional feedback, since the computer knows how many steps the motor has taken and can be programmed to calculate the effect so produced.

The stepper motor requires a repeating sequence of several (usually) 4-bit words to cause it to rotate continuously in one direction which makes it a little more difficult to use than

Listing 4

```
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CA 40 POKE PORT,0
IR 50 FOR STEP=100 TO 200
OD 60 POKE PORT,1+X:POKE PORT,0+X
WC 70 FOR DELAY=1 TO 200-STEP
KI 80 NEXT DELAY
GU 90 NEXT STEP
ZX 100 FOR STEP=1 TO 200
PK 110 POKE PORT,1+X:POKE PORT,0+X
NB 120 NEXT STEP
ZK 130 FOR STEP=1 TO 100
PG 140 POKE PORT,1+X:POKE PORT,0+X
JL 150 FOR DELAY=1 TO STEP
SX 160 NEXT DELAY
NL 170 NEXT STEP
QG 180 IF X=0 THEN X=2:GOTO 50
```

conventional DC motors For this reason, Mullard have developed a chip which greatly simplifies the use of stepper motors, requiring only a single pulse to step the motor and a high/low logic level to determine the direction of rotation A kit containing the IC and external components including a motor, together with a small pcb is available from Maplin Electronic Supplies Ltd, stock code LK76H at the meagre price of £15 95 (Phone 0702-552911 for details) This board will require the simple interface circuit, shown in Figure 5, to convert the 5V TTL pulses from the Atari joystick sockets into 12V pulses required by the motor driver chip

A test program which demonstrates the degree of control obtainable over a stepper motor is shown in Listing 4 the reader should also bear in mind that this circuit is only suitable for uni polar stepper motors If in doubt, Maplin sell a reasonably priced motor that fits the bill nicely, stock code FT73Q price £9 95, but remember this is supplied in the kit if you decide to buy it.

Eidolon

from Activision
XE/XL 64K Disk £14.99

Before embarking on this mystical adventure it is essential to thoroughly read the superb instructions enclosed with the packaging.

As the mystery unfolds, you'll learn that Dr Josef Vincent Agon, an eccentric scientist and inventor, who for many years shut himself away from the world, in his great Victorian Mansion to study the powers of the human mind, mysteriously disappeared without a trace His distraught family closed Agon Mansion and it has remained un-inhabited ever since – that is, until the day when, while out walking, you discover this Mansion, the iron gates ajar, you can squeeze yourself through, you cross the front path cautiously to the front door – the brass knob on the impressive door only needs a touch – it opens – a glow from across the hallway beckons you on and you step slowly towards the room that was once Dr Agon's laboratory As you descend the stairs, you can see a glowing mirror which reflects a spherical machine made of wood and brass, nearby on a desk is a diary – the pages are yellowed with age, you start reading and begin to unravel the mysterious secret behind this strange device You close your eyes and try to imagine what it was like travelling in 'The Eidolon

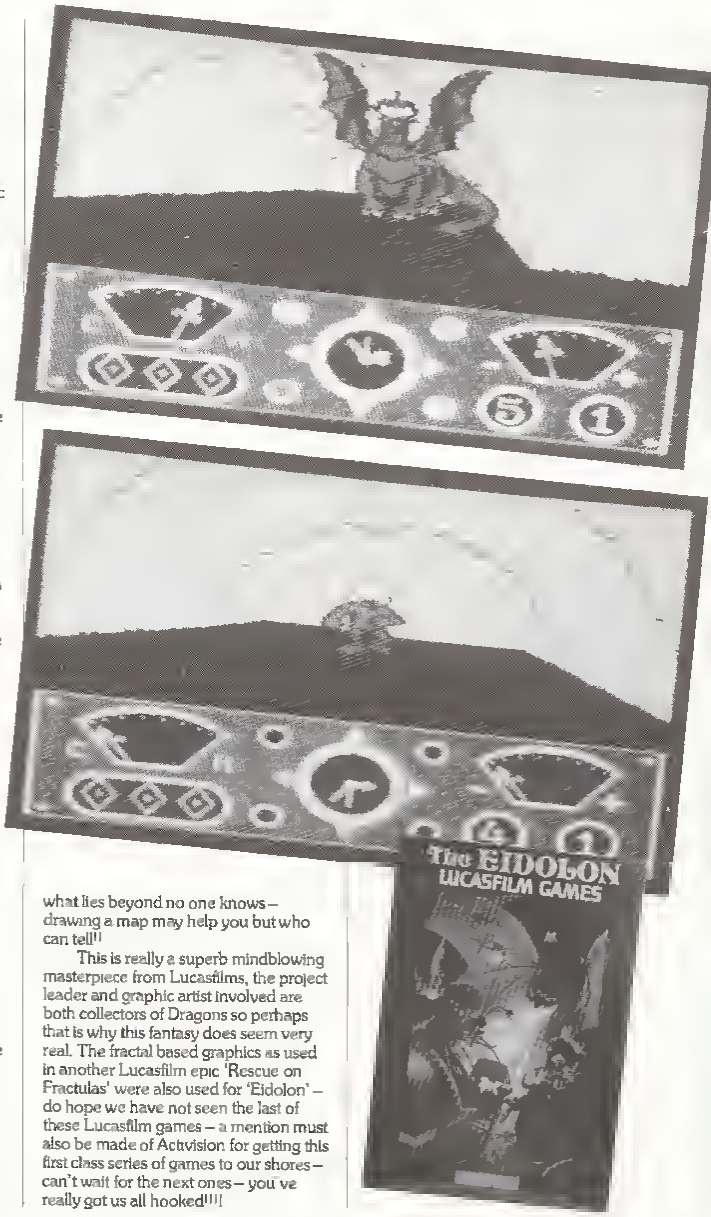
As you read on, something strange begins to happen to you soon you are to become part of the experiment which Dr Agon had written about.

The Eidolon's control stick takes you through many caves as if you are in a dream, walls rush past; the contents of Dr Agon's diary start making sense

Generating coloured fireballs quickly drains the power reserves, so targeting and capturing these fireballs, except the red ones which are destructive, can increase Eidolon's energy reserves Trolls, Puffer Birds, Rotoflies, Biter Birds as well as Dragons of varying sizes and colours Greps and Bottlenecks play their part in 'preying' on your mind!!! Dials help you to track the Dragons and any jewels collected are depicted by coloured diamond shapes which also light up

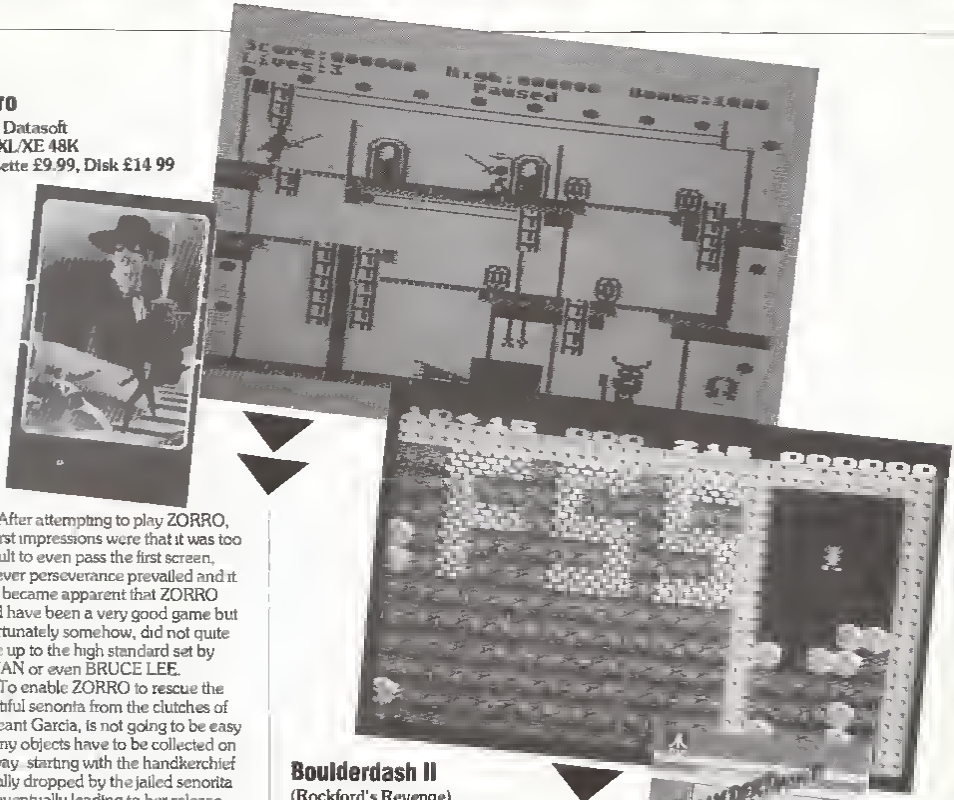
As you become more involved in this mystical world of Dragons, you progress hopfully through to the sixth level, only one more level to go, are you brave enough to proceed remembering what might have happened to Dr Agon, what lies beyond no one knows – drawing a map may help you but who can tell!!

This is really a superb mindblowing masterpiece from Lucasfilms, the project leader and graphic artist involved are both collectors of Dragons so perhaps that is why this fantasy does seem very real. The fractal based graphics as used in another Lucasfilm epic 'Rescue on Fractulas' were also used for 'Eidolon' – do hope we have not seen the last of these Lucasfilm games – a mention must also be made of Activision for getting this first class series of games to our shores – can't wait for the next ones – you've really got us all hooked!!!!



Zorro

from Datasoft
800/XL/XE 48K
Cassette £9.99, Disk £14 99



After attempting to play ZORRO, the first impressions were that it was too difficult to even pass the first screen, however perseverance prevailed and it soon became apparent that ZORRO could have been a very good game but unfortunately somehow, did not quite come up to the high standard set by CONAN or even BRUCE LEE.

To enable ZORRO to rescue the beautiful senonta from the clutches of Sergeant Garcia, is not going to be easy – many objects have to be collected on the way starting with the handkerchief casually dropped by the jailed senorita and eventually leading to her release from her kidnappers

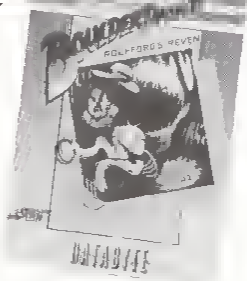
ZORRO also has to solve many puzzles as well as selecting the most useful items on his journey through the various screens

The animation of ZORRO and the guards are well defined – ZORRO bounces on trampolines, jumps off spring boards hangs from a 'look alike curtain rail and wields his mighty sword with great prowess – all the screens are inter-connected so sometimes returning to the hotel room to collect items, such as a branding iron, bell or key which he keeps dropping, or even using a body as a transporter aids his eventual goal – as has been said before, mapping plays quite a part in the rescue of his lady-love

Many strategy minded games players find mastering this type of game a challenge whereas others give up – if you liked BRUCE LEE then you will certainly enjoy ZORRO, so grab your sword and don your mask and get ready to make the mark of 'Z'

Boulderdash II

(Rockford's Revenge)
from Databyte
48K Cassette £9.99, Disk £14 99



Having tried to play BOULDER-DASH 1, and found it exciting but frustratingly difficult to win, I was not looking forward to the Mk. 2 version, with 5 levels of difficulty 324 scrolling screens, 16 all new caves, my confidence sunk very low Having heard comments like, "I've got Boulderdash 1, why do I want another version of the same game" – nothing seemed to be in its favour However this version is so much more complex, strategy and planning once more play a big part in the eventual completion of the game (if ever), that it is like a completely new game Helpful documentation is enclosed so it shouldn't be too difficult!!

Rockford, our 'star', starts by digging to get past the firefly dens, collecting jewels, disposing of the fireflies by getting them to bump into each other, as well as discovering the enchanted walls – by using the Amoeba (a green blob to you) progression through the various caves helps Rockford reach his objective! Each cave is timed, so don't take too long

If you haven't got BOULDER-DASH 1 then I would highly recommend buying this updated version which is very nicely packaged – however if you have ask your friendly ATARI dealer to show you a copy and then you can judge for yourself whether it is worth buying It is also nice to find a game for one or two players, you don't find it that often All in all, an excellent follow up

Whirlinurd

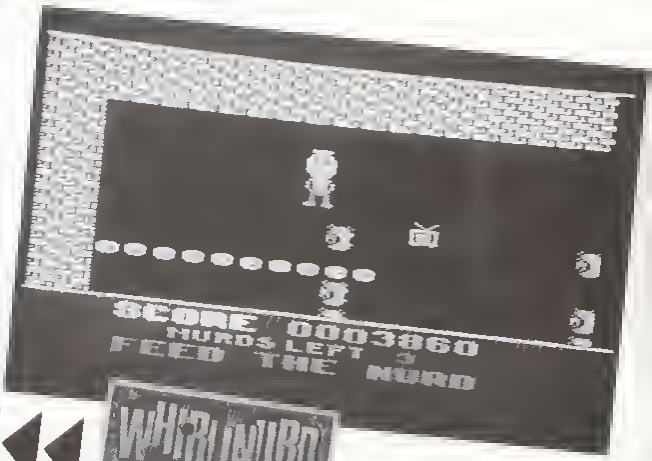
from U.S. Gold
800/XL Cassette £9.99, Disk £14.99

Although there appears to be a bug towards the end of this game, it is still worth reviewing.

The four NURD brothers Haymish Pupkin, Naples Yertz, Hooseby Nurd and Melvin Lugby have run out of food at home so their objective is to visit the greengrocer who unfortunately is out – with the help of a hat made from a propeller they transport themselves around the house looking for something to eat. Collecting food keeps you alive Each Nurd has 3 lives a bonus nurd is awarded after the successful completion of any room

Collecting items such as keys and avoiding slinky snakes, bugs of various descriptions, spudniks and Bouncing baby bilhard balls are all designed to stop you getting the Ultimate Feast.

A nicely packaged fun game – good documentation too



Action Biker

from Mastertronic
Cassette £1.99

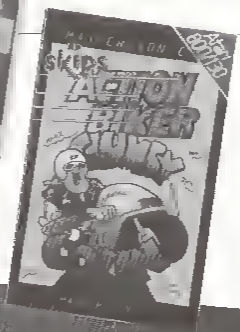
The 'ACTION BIKER' (Clumsy Colin) as he is called is one of the many games on the market from MASTER-TRONIC at an incredibly low price of £1 99 – real pocket money stuff!! It certainly makes a change to be able to review some software that is this reasonable

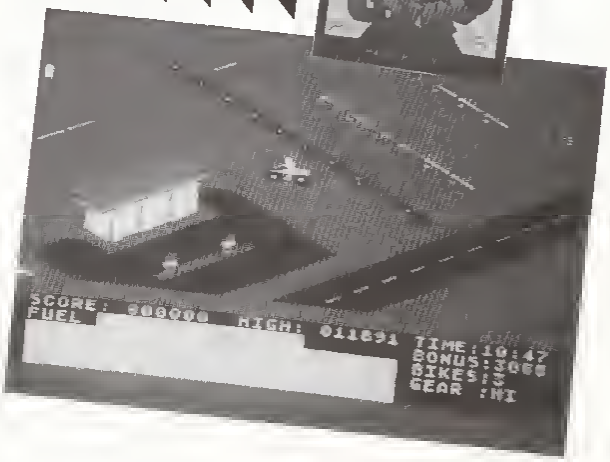
Clumsy Colin has to ride through the streets of the scrolling city collecting items such as a crash helmet and riding leathers, to help him overcome the lake, a fairground and a building site with all its obstacles

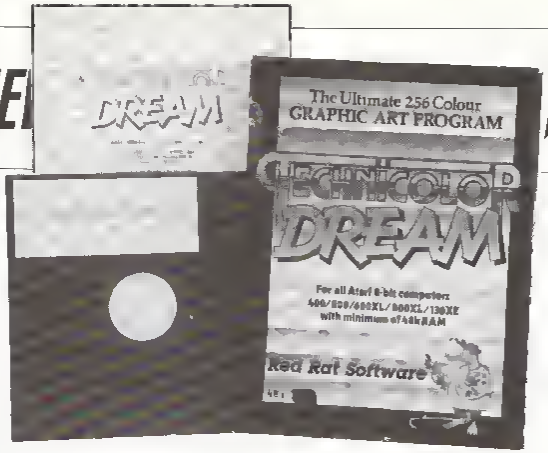
The speed of your journey determines your high score total Your skill in controlling the throttle and gear selection also helps to achieve your total

The game is not so easy as it first looks – if Colin, who's no Barry Sheene reaches the end of the game, then he qualifies for the Drag Race!!

All in all the graphics are quite good, the game fun to play and does become quite addictive, although you will hate to admit it. My only complaint is documentation which could have been better, but for £1 99 who should complain







TECHNICOLOR DREAM

Reviewed by Matthew Tydeman
by Red Rat Software
48k Diskette £12 95
48k Cassette £9 95

Over the years the Atari computer has been well supported by graphics programs MICROPAINTER by Datasoft set a pleasant standard for the Atan computer while the recent introduction of ATARI ARTIST and MICRO ILLUSTRATOR have brought a wel come change, still keeping to a set format and allowing multiple features, that we've all come to know so well.

The past months have brought two new arrivals for artistic Atari owners GRAPHICS ART DEPARTMENT by Databyte being the first of these and one which contains many new and useful drawing commands TECHNICOLOR DREAM by Red Rat Software is the second and is a package which opens up yet another aspect of computer art, drawing with an unlimited palette of 256 colours A difference which is now not exactly unique but effective and saleable. TECHNICOLOR DREAM allows you, the artist and controller, the chance to add a higher level of realism to your artistic constructions, a level which is achieved with an initial range of 256 colours, all of which can be used on one single screen, together with the added bonus of an additional 128 'filters', which can be combined with the selected colours to produce an outstanding colour combination of around 8 million different mixes, as the manual so kindly expresses in its opening introduction. On booting the disk, the programs HELP screen is presented in a rather simple but clear and understandable multicolour graphics mode 0 screen and displays the most commonly used features of the program in the form of CONTROL +KEY combinations This screen is accessible at any time within the program by pressing the ESCape or HELP keys (HELP only operative on machines equipped with this feature – XL/XE only) Colours are selected from a large, screen-sized colour palette of 256 coloured squares Move the joystick (or Touch Tablet Pen) to the colour required and press the START key to acknowledge the colour selected From here the colour pelette screen can be turned off leaving you with a blank drawing canvas The colour palette can be re-accessed at any time for additional colour selection. Colour selection can also be made by pressing the OPTION key for colour and the SELECT for brightness – useful if you know the colour you want and cannot wait for the colour palette to be drawn (some 9 seconds) With your desired colour chosen its onto the constructive part. Drawing with a joystick can be rather hard, especially with a program like TECHNICOLOR, thus the option to support the Touch Tablet Pen is most welcome Movement around the screen is as simple as any other graphics package – if anything, control is rather faster Such features as Colour and Background Luminance, together with a Line mode make a hard job easier

Pictures are saved to a standard DOS disk and appear on the TECHNICOLOR inventory as one visible file which you then select with the load option Files are actually two files – x.LUM and x.COL, which are loaded over the top of each other to get the effect of multicolour realism so cleverly achieved by TECHNICOLOR. To top this, TECHNICOLOR DREAM includes an Advanced Effects section which, when composing your artwork, enables you to alter the colour configuration of your current displayed picture When entering the Advanced Effects mode the actual screen is reduced to a small central black and white rectangle, while your 'Effect prompt' is located beneath this While in this helpful mode one can change as an example, all the brown on the screen to red or green Thus changing of colour will then produce a different effect when laid upon a screen filter or mix The Advanced Effects mode also includes changing of screen contrast, picture negatives, together with changing of colour values. Colours can also be merged for yet further colour combinations

Other features of the program really makes TECHNICOLOR DREAM different from other clones. A store and restore screen (store your picture in memory – if any drastic mistakes are then made the original picture can be loaded from RAM) A resident screen dump program allows instant dumping of your design to any Epson compatible matrix printer at the press of a button On the flip side of the disk you'll find a Basic autoloading graphics handler which allows you to use TECHNI COLOR creations with your programs – something other graphics programs lack (This particular loader program utilises a fast way of loading non-compacted picture files) Printed in the back of the small but sufficient manual are a collection of Basic type in programs allowing conversion of graphics 8 pictures to TECHNICOLOR format and the dumping of TECHNICOLOR screens – this time seperate from the main program and Basic enabling modifications by anyone fluent in Basic programming

TECHNICOLOR DREAM although extremely well featured in the colour and special effect areas does lack many of the standard features even the relatively basic graphics programs include – drawing a circle free hand with a fast cursor isn't exactly a simple task even for someone whose actually got a trained eye in drawing circles on a computer. These simple additions, I feel, would have made the package much more useable and even more powerful – colours are not everything Even though the multicolour graphics of TECHNI-COLOR DREAM are no longer original and unique (Atari's demo disk packaged with the 1050 disk drive contains pictures similar to that of TECHNI-COLOR DREAM which were designed in Germany using a 200 line basic program), TECHNICOLOR DREAM uses a technique which is extremely effective and, if used properly, very professional results can be obtained

TECHNICOLOR has made a good job in establishing Red Rat Software in the Atan market place TECHNICOLOR is a good product as it stands, but a product which would be exceptional if more thought had gone into including those features which are unfortunately missing

THIS SUMMER'S SMASH HIT!

Monthly Hi-Score competition

Screaming Wings

Disk £9.95 Tape £7.95

An all action arcade game. Take off from the US carrier and pilot your plane into battle against the air-force of Japan. This is the South Pacific during World War II and the enemy are coming at you thick and fast. Squeeze hard on your joystick and blast their screaming Zeros from the sky.. Scrolling screen, wild explosive action!

Technicolor Dream

Disk £12.95 Tape £9.95

Unleash the full potential of your Atari computer with this sensational graphic art utility. 256 colours along with 128 filters gives over 8 million possible variations! Demo pictures included.

Panic Express

Disk £5.95 Tape £3.95

The first of RED RAT's budget software. 14 screens of action packed fun.

LOOK OUT FOR SOME EXCITING BRAND NEW RELEASES IN THE COMING MONTHS!!!!

Over 70 Different Attack Waves!

Red Rat Software

15 Fennel Street, Manchester M4 3DU. Tel: 061-834 4941

STARTING FROM BASICS

by Captain Hacker Part Five

Welcome to the fifth part of my series aimed at the beginner.

Arrays

Sometimes, when writing a program, we might want to store a collection of numbers in numeric variables To the beginner, the most obvious way of doing this is to have a separate variable for each number to be stored, i.e A, B, C, D, E Lets invent an example. Suppose that we want a program to store the results of an exam paper given to a class full of students To simplify matters, let us suppose that we have a rather small class of five students, and that we want our program to ask the operator for the marks out of a hundred for each student who took this paper Using single variables for each student, we could write a program such as Program 1

Enter the program and RUN it. Notice how, although it works, it is very cumbersome and rather large, considering the simplicity of the task, and it is of course not very flexible since if we wanted to increase the number of students it catered for wa would have to change the program drastically Just imagine how large the program would be if we wanted to store the results of a hundred or more students!

The key to improving this program must, of course, lie in the way in which we store our numbers in our variables We need to have some way of making a common link in the variables we are storing The answer lies in the humble ARRAY

What is an Array?

An array is a series of numeric variables in which each element does not have its own individual variable name, but instead have a single overall alphabetic name (i e A) So how do we tell each variable apart from the others? Well, to differentiate between each variable in our array (or each element of the array, as they are usually refered to) each variable has its own reference number, starting with zero When we write this we place the reference number in brackets after the array name For example the first variable of array A would be refered to as A(0) and the second as A(1), and the third as A(2), and so on Take a look at Figure 1, and you will see what I mean.

Before we can use an array, however, we need to tell the computer how many elements we will want it to have, so that BASIC can allocate

```
LN 0 REM **** PROGRAM 1 ****
NG 1 REM
HB 100 PRINT "ENTER THE MARKS FOR-"
XO 110 PRINT "STUDENT 1 ";:INPUT A
ZE 120 PRINT "STUDENT 2 ";:INPUT B
BF 130 PRINT "STUDENT 3 ";:INPUT C
DG 140 PRINT "STUDENT 4 ";:INPUT D
FN 150 PRINT "STUDENT 5 ";:INPUT E
TO 160 PRINT
FG 200 REM NOW PRINT THE RESULTS
WP 210 PRINT "STUDENT 1 SCORED ";A
YB 220 PRINT "STUDENT 2 SCORED ";B
AR 230 PRINT "STUDENT 3 SCORED ";C
CS 240 PRINT "STUDENT 4 SCORED ";D
ET 250 PRINT "STUDENT 5 SCORED ";E
RU 260 V=(A+B+C+D+E)/5
TR 270 PRINT
UG 280 PRINT "AVERAGE SCORED = ";V
```

enough space for each number We do this with the DIM command, as in the following example.

10 DIM A(5)

This however is where a lot of people trip up since in the above example we are making room in the array for six variables, not five! This is because arrays have an element zero, (i.e A(0)) You may often find that programs which have a varying quantity of numbers stored in an array will often use elements one and above to store the numbers, and use element zero to store the number of elements used.

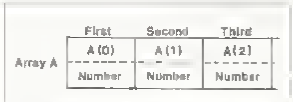


Figure 1

Using our Array

Lets now try and improve our exam results program using an array Type NEW to clear out the old program, and enter Program 2

Notice how much neater and more compact it is RUN the program and you will see that the first thing it asks for is the number of students you have marks for You can of course just enter 5 as in our previous example, but – and here is the real advantage with this method, we can in fact enter any number from 1 to 100, without any changes to the program at all!

You can even extend the number of students to well over a 100 by changing the value of 100 in the DIM command. The only limit to this is how much free memory you have – you can find this out with the the PRINT FRE(0) command, this gives the number of free bytes of memory available to your BASIC program. You should remember though, that each element of an array uses eight bytes of memory

The Matrix

We have just seen an example of a one dimensional number array, where we imagine the array as a row of numbers which are addressed by the overall variable name and a single reference number gives the position for each element. There is, however, a second kind of array called a MATRIX, and here, instead of just one row of numbers, we have many rows of numbers To reference a particular number element in a matrix, we need to specify which row, and which column our element lies in Take a look at Figure 2

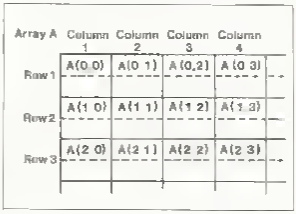


Figure 2

```
RH 0 REM **** PROGRAM 2 ****
NS 1 REM
TV 10 DIM R(100)
XU 20 PRINT "HOW MANY STUDENTS";:INPUT N
HB 100 PRINT "ENTER THE MARKS FOR-"
IE 110 FOR X=1 TO N
PM 120 PRINT "STUDENT ";X;:INPUT M
VB 130 R(X)=M
LT 140 NEXT X
TR 150 PRINT
FG 200 REM NOW PRINT THE RESULTS
IF 210 FOR X=1 TO N
YK 220 PRINT "STUDENT ";X;" SCORED ";
HZ 230 PRINT R(X);:V=V+R(X)
LU 240 NEXT X
TR 250 PRINT
UY 260 PRINT "AVERAGE SCORED = ";V/N
```

By now, some of you (particularly those whose maths is not very strong!) may be rather confused, so lets try and simplify things a little by forgetting about computers for a while, and instead relating the concept of arrays and matrices to something more real! Lets call upon our pots, (or jamjars, if you like) once again.

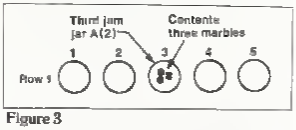


Figure 3

Suppose that we have a row of five jamjars, inside of which may be any number of marbles. If we want to tell someone to put three marbles into a particular jar, but without pointing to it, we would probably say something like "The third jamjar along" this is precisely what happens in our first array, we would write this in our program as A(2), (remembering of course that we count from zero) Figure 3 shows what I mean.

So, what about our matrix? Well, take a look at Figure 4. Now we have four rows of jars, with five jars in each row. If we want to tell someone to put three marbles into a particular jar it is now no longer sufficient to say to them "The third jar along the row", because they would immediately ask "which row?". We must now of course say which row, and which column a particular jar is in – in our case we would say "Row 2 and column 3". We must do precisely the same thing with number arrays, so our little jamjar example would be A(1,2), once again remembering that element numbers start with zero. Therefore we might write the following program line.

130 A(1,2)=3

Using the Matrix

We have seen that the matrix is simply a two dimensional version of the array, but what about using it in a program. Remember our student's exam result routine (Program 1)? Well

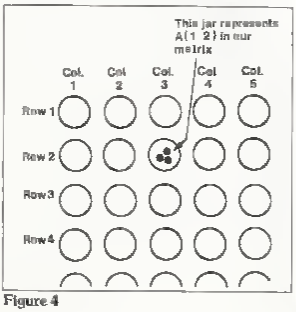


Figure 4

```
NG 0 REM **** PROGRAM 3 ****
NG 1 REM
XE 10 DIM A(100,20)
HL 20 PRINT "HOW MANY STUDENTS";:INPUT N
ZI 30 PRINT "NO. OF TESTS EACH";:INPUT T
HG 100 PRINT "ENTER THE MARKS FOR-"
IE 110 FOR X=1 TO N
NN 120 FOR Y=1 TO T
VS 130 PRINT "STUDENT ";X;" - TEST ";Y;
ZC 140 INPUT M:A(X,Y)=M
QR 150 NEXT Y:PRINT
LX 160 NEXT X
TQ 170 PRINT
FG 200 REM NOW PRINT THE RESULTS
IF 210 FOR X=1 TO N
PG 220 PRINT "STUDENT ";X;" SCORED-"
HK 230 FOR Y=1 TO T
FW 240 PRINT A(X,Y);" IN TEST ";Y
LN 250 V=V+A(X,Y)
QU 260 NEXT Y:PRINT
NA 270 NEXT X
TT 280 PRINT
EU 290 PRINT "AVERAGE SCORE = ";V/N/T
```

suppose that each student's exam consists of 4 different papers, once again with marks out of 100 for each paper. Using a matrix this task becomes much simpler, as you will see in Program 3.

ASCII

You will probably by now have seen the terms "ASCII" and "CHARACTER SET" mentioned quite regularly (particularly if you are a regular magazine reader), so for those of you who are not certain about their meanings I shall try to explain.

Firstly, though, you will need to know how a string is stored in a computer. Your computers memory is composed of a collection of locations which can hold a number of values between 0 and 255. These locations are refered to as BYTES (and believe it or not, a NIBBLE is half a BYTE!), and when BASIC holds a string of characters in memory it places each character into a byte of its own, one after the other. Now, to do this it must of course give each letter a value, or number, with which it can be refered to. This is where the ASCII character set comes in.

The letter A is actually stored inside your computer as the number 65, and B is stored as 66. C is stored as, you guessed it, 67. This sequence continues right through to Z, which is character number 90. Bearing this in mind, if you were to store the word "CAB" in, say A$, you would be placing the numbers 67, 65 and 66 into three consecutive locations somewhere in memory.

So how can we find out the 'values' of characters in a string? The answer is to use the ASC command. This command will convert a character into a number to store in a numeric variable or to print to

```
NV 0 REM **** PROGRAM 4 ****
NG 1 REM
OZ 10 DIM A$(100)
AV 20 PRINT "ENTER A CHARACTER...";
YC 30 INPUT A$
SH 40 PRINT ASC(A$)
AV 50 GOTO 20
```

the screen. Enter and RUN Program 4.

Enter any letter from A to Z, and press RETURN. The program prints the ASCII value for your letter on the next line. The ASC command doesn't have to be used with a string variable, you can actually use a literal string.

PRINT ASC("B")

This will print the value 66. Notice that you must put quotes around the character.

Suppose, however, that we use a string of two or more characters in this command, what happens then? Try the following line.

PRINT ASC("HELLO")

Notice that the ASC command has only taken notice of the first character, printing its ASCII value of 72, and ignored the rest. The next question is bound to be how can we get the ASCII values of the other characters in our string? We must use the 'part of string' facility, for example

PRINT ASC(A$(3,3))

This will print the ASCII value of the third character in A$. Now type in and run Program 5. Notice that Program 5 introduces another new function called LEN (on line 40), this function gives us the real length of A$, which is of course not always the same as the DIM length of the string.

Lower Case and Numerics

Whilst you have Program 5 running, try entering some numeric characters, 0 to 9. Notice that each of these digits also has an ASCII character, ranging from 48 for "0", to 57 for character "9". At this point you should appreciate that numbers held this way (in a string variable) are completely different to those held in a numeric variable (where they are stored in an encoded 6 byte form). Punctuation symbols also have their own numbers, as do lower case letters. Lower case letters range from ASCII 97 for "a" to

```
OP 0 REM **** PROGRAM 5 ****
NG 1 REM
OZ 10 DIM A$(100)
JT 20 PRINT "ENTER YOUR STRING...";
YC 30 INPUT A$
LN 40 FOR B=1 TO LEN(A$)
KF 50 PRINT A$(B,B);" = ";
NG 60 PRINT ASC(A$(B,B))
SK 70 NEXT B
CJ 80 PRINT :GOTO 20
```

DECIMAL CODE	CHARACTER	DECIMAL CODE	CHARACTER	DECIMAL CODE	CHARACTER	DECIMAL CODE	CHARACTER	DECIMAL CODE	CHARACTER
0		26		52	4	78	N	103	g
1		27		53	5	79	O	104	h
2		28		54	6	80	P	105	i
3		29		55	7	81	Q	106	j
4		30		56	8	82	R	107	k
5		31		57	9	83	S	108	l
6		32	Space	58	:	84	T	109	m
7		33	!	59	;	85	U	110	n
8		34	"	60	<	86	V	111	o
9		35	#	61	=	87	W	112	p
10		36	$	62	>	88	X	113	q
11		37	%	63	?	89	Y	114	r
12		38	&	64	@	90	Z	115	s
13		39	'	65	A	91	[	116	t
14		40	(	66	B	92	\	117	u
15		41	)	67	C	93	]	118	v
16		42	*	68	D	94	^	119	w
17		43	+	69	E	95	–	120	x
18		44	,	70	F	96		121	y
19		45	-	71	G	97	a	122	z
20		46	.	72	H	98	b	123	
21		47	/	73	I	99	c	124	|
22		48	0	74	J	100	d	125	
23		49	1	75	K	101	e	126	
24		50	2	76	L	102	f	127	
25		51	3	77	M				

Here is the Atari's Character Set. Characters 128 – 255 are the inverse-video versions of these *except 155*, which is the Carriage Return code

Table 1

ASCII 122 for "z"

These values are called the standard ASCII character set, and you will find that most computers will have the same numbering format for these characters. There are, however, other special characters on your computer which are unique to the Atari in their function and numbering format, and you may sometimes hear books refer to the Ateri version as ATASCII. Table 1 shows the Atari character set.

Producing Characters

Well, now you have seen how to look at the ASCII value of any character in a string, but what if you want to do the opposite, and produce a character using its ASCII value? Well, you can do this using the CHR$ command. Type the following line, and press RETURN

```
PRINT CHR$(65)
```

You will see that the computer prints the letter A. The CHR$ commend can also be used to insert characters into a string, or even to build a string. Try the following program, and you will see what I mean.

```
10 DIM A$(3)
20 A$="PIT"
30 A$(2,2)=CHR$(65)
40 PRINT A$
```

You will see that the word PIT has been changed to PAT.

Some Strange Characters

Let's take a look now at some of the characters that are special to the Atari. Enter the following program and then RUN it.

```
10 FOR A=0 TO 26
20 PRINT CHR$(A)," ",
30 NEXT A
```

You will see a row of strange symbols on your screen, these are the graphics characters which you can also obtain by holding down the CONTROL key and pressing any of the A to Z keys, or the comma (,) key, these characters can be used to create pictures on your screen, and you can either print CHR$'s in your program, or by inserting the control A to Z characters into strings directly from the keyboard.

Special Function Characters

Some of the characters in the Ateri character set are screen control characters, i.e. rather than print a symbol on the screen, they will normally have some effect on the contents of the screen, or upon the line that the cursor is on. Perhaps the most dramatic effect is produced by character number 125. Try the following line.

```
PRINT CHR$(125)
```

You should have noticed that the screen has cleared, just as if you had held down SHIFT and pressed the CLEAR key!

This is in fact how the clear key (and all the other screen and cursor control keys) work. They cause the computer to print the relevant control character to the screen immediately. Try this one.

```
PRINT CHR$(253)
```

You should have heard the keyboard buzzer buzz! (If you have an XL or XE computer this would go through your TV speaker.) Even the cursor move keys (up, down, left and right by one position) operate via control characters, and can be used in a program, as the following example shows. Enter it into your computer, and before you type RUN, press the return key until the cursor is at the bottom of the screen.

```
10 FOR A=1 to 15
20 PRINT CHR$(28),
30 NEXT A
```

You should see the cursor jump up towards the top of the screen, just as if you pressed the 'cursor up' key combination 15 times!

Now Let's 'Escape'

Now you have seen the effects of some of the screen control characters, but these characters do actually have symbols which can be printed onto the screen. To see these we have to somehow tell the computer not to carry out the characters function, but to just print its symbol to the screen and treat it like any other character or letter. We can do this with the 'ESCAPE' character. Type in the following line and press RETURN.

```
PRINT CHR$(27),CHR$(125),
```

Notice that, this time, instead of clearing the screen it prints a little arrow. The ESCAPE character (CHR$(27)) told the computer to ignore the control function of the following character, and print its symbol instead. Try each of the following as well.

```
PRINT CHR$(27),CHR$(253)

PRINT CHR$(27),CHR$(28),
CHR$(27),CHR$(29)
```

Notice however that the escape character only works for the one character which immediately follows it.

Even the ESCAPE character itself has a printable symbol, as the following line will demonstrate.

```
PRINT CHR$(27),CHR$(27)
```

Take a look at the top left hand key on your keyboard, the one marked ESC. This is the key which generates the ESCAPE character. To see how this works, type the following line but DON'T press RETURN.

```
PRINT "
```

Now press the escape key once. You should not see anything happen on the screen. Now hold down SHIFT and press the CLEAR key. Usually this key will cause the screen to clear, but because we pressed ESCAPE the clear character has been stored like a normal letter, and its symbol has been printed! You can now enter the closing routine ("), and press RETURN. Now of course, the computer clears the screen because the ESCAPE code was not actually been stored in the string, just the clear screen code. If we had pressed the escape key twice, however, the escape code would itself be stored in our string, and we would have to press the ESCAPE key a third time to enable us to enter the clear screen character. Try this, and you should see that the symbol is printed instead of cleaning the screen. Experiment with the ESCAPE key, and try using it in Program 5 to print the ASCII values of the control characters. Remember that there are many more control characters than the few I have mentioned so far, such as insert character, delete character, insert line, delete line, tab, etc.

There is in fact another way to disable control characters, and this is to POKE 766,1. The effect of this is the same as with the ESCAPE character, but this time the change is more permanent. The control characters will stay disabled until you either press the BREAK key, or you POKE 766,0. This is particularly useful if you are examining memory, or are dumping a file to the screen, since the randomly occuring control codes will not disrupt or destroy the screen display contents.

Inverse Video

Some of you may be wondering how the ATARI produces inverse video text. It is actually quite simple, for inverse video characters are simply the normal character values but with 128 added to them. For example, to print the letter 'A' on the screen in inverse video you might do the following:

```
PRINT CHR$(65+128) or
PRINT CHR$(193)
```

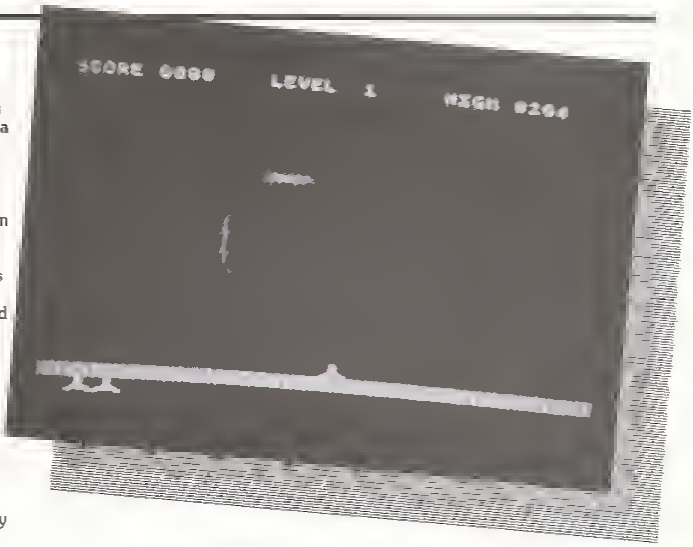
Enter and RUN the following program, which should demonstrate what I mean.

```
DZ 10 DIM A$(100)
YB 20 INPUT A$
LE 30 FOR A=1 TO LEN(A$)
VN 40 B=ASC(A$(A,A))
UH 50 A$(A,A)=CHR$(B+128)
GA 60 NEXT A
MP 70 PRINT A$
CJ 80 PRINT :GOTO 20
```

Enter a string, and the program will convert it to inverse video and print it.

Well that's all for this episode of 'Starting from Basics', see you all in the next issue.

Another Boring Space Invaders Game



by Bob Askew - Northampton
Runs in 32K Cassette or Disk

Contrary to the title of this game it is actually very good, easy to play and a lot of fun. Type in the program, if using a checksum program such as our own KEYO or something like TYPO, utilise the two checking letters preceding each line. If you are not, then please ignore the first two characters that are printed in front of each line number. Once you have completed typing in the listing save a copy to your disk or cassette, this is just a precaution in case there is a problem when you run the program and all of your typing is lost.

Type RUN and press RETURN, you will be prompted to type 0 or 1 depending on whether you wish to create an auto-boot cassette or disk. After this the program will check itself line by line. When completed an Autorun Sys File is written to your disk, or an Autoboot File is saved to your cassette. So to play the game, just autoboot your disk or cassette and away you go.

```
FK 10 REM ANOTHER BORING SPACE INVADER GA
   ME
UH 20 TRAP 20:? "MAKE CASSETTE (0) OR DIS
   K (1)?";:INPUT DSK:IF DSK>1 THEN 20
UK 30 TRAP 40000:DATA 0,1,2,3,4,5,6,7,8,9
   ,0,0,0,0,0,0,0,10,11,12,13,14,15
BK 40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
   READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
   E 1000:TRAP 120:? "CHEKING DATA"
JB 50 LINE=LINE+10:? "LINE ";LINE:READ DA
   T$:IF LEN(DAT$)<>90 THEN 220
ZY 60 DATLINE=PEEK(193)+256*PEEK(194):IF
   DATLINE<>LINE THEN ? "LINE ";LINE;"MIS
   SING!":END
LY 70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
   X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
   EX(D1)*16+HEX(D2)
FG 80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
   EAD CHKSUN:GOTO 50
BG 90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
    TOTAL=TOTAL-1000
LR 100 NEXT X:READ CHKSUN:IF TOTAL=CHKSUM
    THEN 50
MO 110 GOTO 220
ZN 120 IF PEEK(195)<>6 THEN 220
EB 130 IF PASS=0 THEN 170
NY 140 IF  NOT DSK THEN 160
KS 150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
    #1,2:PUT #1,0:PUT #1,56:CLOSE #1:END
   :? #2
IJ 160 CLOSE #1:END
IG 170 IF  NOT DSK THEN 200
BD 180 ? "INSERT DISK WITH DOS, PRESS RET
   URN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0
   ,"D:AUTORUN.SYS"
ZH 190 PUT #1,255:PUT #1,255:PUT #1,0:PUT
    #1,55:PUT #1,23:PUT #1,76:GOTO 210
BD 200 ? "READY CASSETTE AND PRESS RETURN
   ":OPEN #1,8,128,"C:":RESTORE 230:FOR
   X=1 TO 40:READ N:PUT #1,N:NEXT X
QS 210 ? :? "WAITING FILE":PASS=2:LINE=99
   0:RESTORE 1000:TRAP 120:GOTO 50
NI 220 ? "BAD DATA: LINE ";LINE:END
NC 230 DATA 0,43,216,54,255,54,169,0,141,
   68,2,169,60,141,2,211,169,0,141,231,2,
   133,14,169,76,141,232,2
KX 240 DATA 133,15,169,0,133,10,169,56,13
   3,11,24,96
ET 1000 DATA 707070707047237770784737377
   7047483777D7D7D7D465F377D7D7D742733741
   0037000000212E2F342825320,328
DT 1010 DATA 00222F32292E270000000000033
   3821232500002922E342124253200000000000
   0000000027212D250000000,240
ZK 1020 DATA 000000000000000000A2890DA2AF
   A200A103A0A5B7D0000000000000000000000
   0000000000000000000000000,945
ML 1030 DATA 0000000000000000000000000000
   0000000000000000000037245737300337461727
   7400216E640D2265D0226F72,549
YA 1040 DATA 656400346F00346561727301D0D0
   0064D0D645A64D0D96D0D96BC960D0D064640D64
   64D0D05A5A0D0545400D054D0,225
GE 1050 DATA 4E4E0D0D0D0D0D0D0D0D0D0D0D0D
   0000000000000000000000000000000D0D0A903
   0D0FD2A9738D1E37A90D0DC5,936
FH 1060 DATA 0285CB85CC85CD8D300D2A937BD31
   02A943BDC4029DCB02A9EA8DC402A9A98D0C4B2
   A9A6D0D1D220603020603B20,819
NS 1070 DATA A03020603BA91485CDA9CB85CEA2
   20EE1E3720603BA5CE8D0B0230E9058SCECA0B
   ED4C7538E4CB4A4C8D0FAE4CC,42
BD 1080 DATA A4CCC4CDD0F2A0B084CB84CC60A9
   000D0BD285CDAD1FDAC9D6D0F9A9AABD01D2A9
   A46D03D2A200A9460SCD2003,450
ZF 1090 DATA 3020603BE0D0D0F6A91E85CD2003
   3020603B9EB21DBF6A9D0BD0D2D24CC33B00C337
   6D0D02F00638E90206D2E0,641
ON 1100 DATA 602B603B206D3B4CD03900000000
   0000000000000000000000000000000000000000
   000000000000000000000000,438
EH 1110 DATA 0000000000000000000000000000
   00000A9000580BD2F02BD143FBD403FA8AABD
   30D2A93FBD31D2A9BABD413F,999
GM 1120 DATA A90CB0C502A94E8DC202BDC3D2A9
   9D8DC0D2BDC6D2A9CCBDCD02BDC4D2A9500581
   A9009D0006E8E0D2FD0F0A200,498
DL 1130 DATA 9584E8E0160DF9A2009180C0D0F9
   E6B1E8E0D60F4A24CB6B1A8E0B4830EF402A9
   008502AAA801829180C8D0F9,15
```

```
JR 1140 DATA E6B1E6B3EBE0D2DDF0A2008D9C39
9D1D4CEBED0DDF5A0DD89A439991D4CC8C008
D0F54CAC3918181818183C7E,750
IG 1150 DATA C3B1812424181B4242A9FA85CCA9
508D07D4A9D3BD19D60D1DD0A2009D7C3FEBE0
03D0FBA91BA2049D313FCAD0,993
OY 1160 DATA FAA93E8D2F02A9808DC756A9D18D
C757858E8D2306A91EB5B0A92885B98D2806A9
A08D1B06A9D98D1806AC803A,985
IE 1170 DATA 0000B0A9538D233FA92885D6A244
A800A9072D5CE4A9B0D514A514C932D0FA20D5
3620863CAD2586F0034C403E,293
HZ 1180 DATA AD1D86F00320863CA50CF00849FF
4AAABDB20AA90885BC2CF73AA0D7B6D02CA9D2
85CD6B0D706AD0AD229FD090A,869
UH 1190 DATA BDC1D2AD0AD229D7AABD6A3AA020
E43A4C753AD0102D3D405B6B7D4C223AA5DDF0
0320D63AA3BDFB032B033B2D,355
LK 12D0 DATA 0B3A2D3B3AA514C9D1D00320A43B
AD7B02C98C8017C9B5B00A5864C93BF0CFD684
4CB03AA584C9C8F0C4E684A5,2B1
MV 1210 DATA B45D0D0D4C723AA9088D0B6EED0
86AD0806D0F86D18A58E65B665B0658DD012A5
B9B58618A58B49B685B78D05,426
IC 1220 DATA DDA9DB85CD6DA2B069D0429D0806
C8E8E010D0F4A9B858A60A200EED106AD0106
C9DAD032BED1D6ADD2D6FD4A,269
HC 1230 DATA AD1BB6C9ADD0DAA9AABD1B86A90A
BDD0D2CED20AEED6A6AD0A06C98AF0034C5530
8E8606EE0506AD05D6C9DAFD,452
UO 1240 DATA 034C55308E0506EE0406EE1CD6AD
B4B4C98AF0034C5530E0406EED3D619D0306
6910D323FE8E0D4D0F2AD1B,324
XK 1250 DATA B6C9ADD00A9008D0DD24C7638CE
1BD6AD1B06BD01D2AC1A06D9BE3BDD1CB6D003
8D1D06A9DD58E6D0D02D01,265
LK 1260 DATA 01010101010201D1D1D1D18101
01D1D181B4D23866D24B6D01FAD1A06C90980
18AD0AD2C907D011AAA94BBD,893
BD 1270 DATA 153FBDCD38BD143FA91E8D2306A0
00265078A0CB5008A5C4D0FC18AD23066D24B6
DD29A94ABD153FADBAD229DF,187
GE 1280 DATA AA6DF838BD143FA91B8D23B64CDB
3CDB285D7BADCB285D5B7BA0CBA02850A0CE19
B6AC19B6CDFFD0034C963D89,869
CR 1290 DATA 833C8D00D0858A4A900997C3FAA9D
CB54E8E008DDFBA9A4BD81D2A95ABD1E06AC10
06A2DBBD1B4C99D954CBE8ED,480
TG 1300 DATA BBD0F4A9809D054B514A514C9D3
98FACE1BD6AD1ED649FF8D1ED68D00D2AD1006
C9C9D8CDA9DDDDDD2997C3F,466
JV 1310 DATA AAA9C4BD01D2B614A514C9D19DFA
E6B43BA5B449FFE9789D0D2A5B4BD0DDC968
D0DEA9D9BD1BD6A92BB5B9A9,840
XU 1320 DATA 1EB5BBA9ABBD01D2A9009B8D85BE
A9B185CD6B384B4A93C8D1FB4EE1A06EE2206
AC1AB4CD15DD06BC25B4C41,388
JY 1330 DATA 3D09423DA5CBB9573D85C9B96C3D
85CA8913DB5C8AD2206C90AD012A9D08D2206
EE21B41BAD21B669188D4D3F,43
PZ 1340 DATA 9B1BAD22B6691B9DA13FA9008D1C
B69D1DD618AD23B66D2406DD2CAD9AD2C946BD
25A9498D153FAD1AB64AABD,985
GU 1350 DATA 343080143FA9018D23B64C413D20
285D5B7B79A8A8ADC8C8080868081B1820201
020201020101D1D2D2D2D201,338
VD 1360 DATA B1D2D2DDDDDD8D8D2B1020102006
868810102020BB2D1D2B0000000000000000
DDD1818181B1D1D2D2D2D2,746
VW 1370 DATA 80D2D4D2D4B4D2D4B4B4D2B4B4D2
B484D4B2D4D4B4A2BDAD4C3FCD323F901DF803
4CC83DAD4D3FCD333F9D1BFD,399
DB 1380 DATA B34CC83DAD4E3FCD343F90834CC8
3D6D323F9D4C3FEBED4D0F5A2DCA98ABD2102
A98FBD233F8D80D2E820353E,690
YM 1390 DATA A9538D233FE820353EA9008D0002
E000D0E28ED1D2A98F8D233FA9018C5C6A98A85
C745C68DC5D2BDC6D2BDCBD2,169
JE 14A0 DATA D0F3A9142037E4A9288D143FA940
BD153FA90E80C502A9142037E4B6E8A90D6D25
868D143FA9E8D0F4824C0036,416
LH 1410 DATA A9BBB514A514C904D0FA60A53B0D
FCA20B6E1FD6BD323F9D4C3FEBE004DDF5A9D1
858DA9008D08D09D04D8BD02,654
XN 1420 DATA B6A95B6D143FA94B8D153FA9A86D
01D2A9C4B0D502A9A28D03D2A95ABD02D2A2C8
6A8D0BD23GE9D5BD04D2BDC5,349
RC 1430 DATA 02AA2D353EE00288EBA900B0D0D2
8B01D2BD02D28D03D28D04D28D05D2A90E80C5
D22D373EA97B8D143FA9142D,1
AD 1440 DATA 373E4C253EBDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,293
MG 1450 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDD7D7D422B3FBD44BF3FD6
B6B6B6B6B6B6B6B6B6B6428B4B,183
KH 1460 DATA 466C4DD6D6D6D6D6D6D6D6D64453
3F4B7B3F41BD3FBD33232F3225D81D1D1D1B8B
DDDDD2C2534252CBDDDBADD,567
JI 1470 DATA DDBDDDDD2B29272BDDDDDDDDDDDD
DDDD2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2D2
D2D2D2D2D2D2D2D2D2D2D2D2,795
HU 1480 DATA D2D2D2D2D2D2D2D2D2D2D2D6D3D3
B3DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,814
CD 1490 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BE 14B0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BH 14C0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BK 14D0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BN 14E0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BQ 14F0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BT 1500 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BW 1510 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
BZ 1520 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CC 1530 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CF 1540 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CH 1550 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CJ 1560 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CM 1570 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CP 1580 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CS 1590 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CV 15A0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
CY 15B0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DB 15C0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DE 15D0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DH 15E0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DK 15F0 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DN 1600 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DQ 1610 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD,816
DT 1620 DATA DDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDD42247E5A7E5A42
4242247E5A7E5AB181183C7E,4B4
UY 1630 DATA A8AB7E3C181B3C7E0507E3C1824
243C5AFF7E3CDB24243C5AFF663CDB2826810
7CD6FE7CB2442B17CD6FE7C,6
WA 1640 DATA FE1B1B1B107C547C3844921078C54
7C9B0899D08FF243C0BBB1B3C5AFF45BDB18B
1C3E6B3E1D2A410B0000000B,500
ZE 1650 DATA 0000B102B461E1E462B1B0B0FF
0F0FFF0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
YD 1660 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
YG 1670 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
YJ 1680 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
YM 1690 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
XN 1700 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
XQ 1710 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
XT 1720 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B,6DB
WH 1730 DATA 0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0B0B
0B0B0B0B0B0B0B0B0B0B0AC10,796
HK 1740 DATA 4CA2B18D1B4C9D0F4CE8E0B8D0F5
BC174CAD2306F010EE2306AD2306DD186D143F
A901B0240BAD240BF003EE24,216
NP 1750 DATA B6AD1F06F003200A46AD9BDBFD15
8588A435848CA98C95C449B685C72DFC44A900
85D5AD09D02901F01065B0D5,854
WA 1760 DATA C420A344A900850B8D04D085B5A5
C4FDD320FE45A5C4F0D320F746A5BEFD034CA1
45A5BBF003206045A406F059,743
```

```
SK 1770 DATA 2DA3442BD244A5B6C9D4904D2BA3
44A9018E5CDA7B9E5E64CE244A7BDAAA4B69980
53C8E9E0B8D0F7601BA5B665,498
BD 1780 DATA CB85B6A8A904A2DB990853C8E8E0
B8D0F7A5CAF011A5B7C5B4B00765CA85074CDC
44E5CAB5B7A5B7BD05D86BA5,99
NE 1790 DATA 85F04B2AFC442D0F45A5B5C926B0
4B2DFC44A9008585C42645A2004485B9005329
B4990053C8E8E0B5D0F26038,222
YT 1800 DATA A5D5E9038585A6A2D5B9005389B1
990053C8CADDF46018BAD840265BD00CA9C485
B518A5B469028DB4D0A58BD0,824
JU 1810 DATA 63A9558SC5FA5B985CEA5BAAAA58D
D0151BA5CC49FF658A691E38ED1AB6BD02D2A9
C1BD03D2E6CCD08AA0FA84CC,245
LV 1820 DATA A5BA4918D5DAA0088DDB8491CEE0
C8C0B9DBF5A5B6C9DC9D1420A745A5BDFB02B5
BE18A5B965C985B9A91E9586,434
UO 1830 DATA 1BA5B865C885B8DD1DBBD1ED04C
62E4A9DB854DAAA489998B55C8E8E008D0F760
2BA745A208BE86D2BE07D218,79
IF 1840 DATA A589A9038589A8C9D9015B62B8E
07B6BE05D0A922B5B9A91E85D685BE4CFD458D
08D699B055C8E8E008D0F4AD,921
ID 1850 DATA 8CD0F008A9018SBD85C4858748A5
BFD02DA5B485C885C1A9C285C2A9CC85C3858F
A901B5C685848DD0D0A9ABBD,575
8U 1860 DATA 18D6BDD1D2A99FB5C7A900BD08D2
BDD2B6E6C5A5CD29FCFDD4C6CBC6DA5C129FC
F004E4C1E6C1C6C2C6C2A5C3,256
PR 1870 DATA 29FCF004E6C3E6C3A2DA4C2B092
469900569900857C8E8E8D5D0F1A2D84C36D92
469900D5699DB57B8E8E8D5D0,367
YS 1880 DATA F1A5CB8D02D8A5C1BDD3D0A5C5C9
7ED008A900B5C585C4858F6008008008E4C6
E4C6A5C6C9B4900EC6C7A5C7,722
CV 1890 DATA C900D006A900B5C685C7A5C7BDD3
D2A5C68D02D268A9C4BDD5D2CE1F06AD200688
D84B249FF8D2806AD413FC9B8,177
ZC 18A0 DATA 90B8A9DABD413F4CE54618AD22B6
691B8D413FAD1F06D0111BAD22D6691B8D413F
A9808D842B805D2600000000,817
FP 1910 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
FS 1920 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
FV 1930 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
FY 1940 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
GB 1950 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
GE 1960 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,817
JT 1970 DATA 0000000000000000000000000000
0000000000000000000000000029340733
0021D02229340332C2F3700,325
IG 1980 DATA 212E2400222F32292E2700293332E
B7340D293401000000000000000000000000
2132250039F23580222F3225,433
GN 1990 DATA 240039253412F0000000000000000
0000000000000000000000000000000003729
213400210039213720E01D000,51
WD 2000 DATA 0000000000000000000000000000
0000000000000000000000222F32292E27
01010100000000000000000,311
SC 2010 DATA 0000000000000000000000000000
0000000342829330029330342F2F00252133
39010000000000000000000,912
DZ 2020 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,912
PE 2030 DATA 0000000000000000000000000000
0000000000000000000000222527292E2532
33002C3523280100000000,423
PY 2040 DATA 0000000000000000000000000000
29072C2800D2128252029340028213242532
00262F32003F35B000000,348
MW 2050 DATA 0000000000000000000392F3500
21292E340033252500E2F342829E27003925
34010000000000000000000,260
CH 2060 DATA 0000000000000027253434292E
27002821322425320033342926200000000000
0000000000000000000000,44
LL 2070 DATA 000000372F37010029334300282F
3434292E27003530010000000000000000000
0000000200000000000000,765
ND 2080 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,765
01 2090 DATA 0000000000000000343239002I
2E2400232F2E23252E343221342501000000000
0000000000000003000000,510
KY 21A0 DATA 0000000342821340037213300302
21342825342923B1B1D0000000000000000000
0000000000000000000002E,212
HJ 21B0 DATA 2F37002F30252E00392F35320025
3925330100000000000000000002F302532
213425003420250033342923,494
IE 21C0 DATA 2B003729342B002B212E24330E0E
2E2F34002625253491D0000000000003002621
0100282101003325323625330,697
EG 21D0 DATA 0B392F3500322972B34010000000
0000000000000000000000000000000000000000
0020000000000000000000,77
DZ 21E0 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000392F,181
TN 21F0 DATA 35B732250024252124010100000000
0000000000000000000000000000232F2E27
322134352C2134292F2E33D1,142
BE 2200 DATA 0029002120D0024252625213425240
01000000000372522C2B037282134002422F00
392F350025383D2523341F0D,464
TD 2210 DATA 2102222C2F2F2439002D2524212C
1F0000000000000000000000000000000000000000
0020000000000000000000,988
PM 2220 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,988
PP 2230 DATA 0000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000,988
```

HOT GOSSIP

•U S GOLD – nice to know some Software Houses keep their promises, a copy of their Bulletin has arrived, not a lot scheduled yet for ATARI, one or two adventure games but as they seem to have their fingers in so many pies, there does always seem something new around with their name somewhere on the packaging

•QUESTPROBE featuring HUMAN TORCH and the THING by Scott Adams is now in your shops price for Disk only £14 95 – 130XE owners get an advantage with this particular piece of software because of their increased K It looks very good and is third in the series written by Scott Adams.

•Still on the subject of Scott Adams The Official Scott Adam s Adventure Hint Book – a User s Guide – price £2 99 is now available for all you adventuring maniacs

•BACK TO THE FUTURE – unfortunately Electric Dreams will not be releasing this for the ATARI and at the moment have nothing scheduled for that machine – pity

•PAPER BOY and BOMB JACK from Elite will also not be available for the ATARI, and here again nothing planned

•TOMAHAWK – Digital Intergration – no news yet, still may do in 6 months time

•MARTECH, despite a release date of 26th May, still have no definite news of ZOIDS – May is still a possibilty so we keep being told but I'm afraid this is not good enough – nothing else otherwise

•TWISTER from System Three Software has been released, that is according to their advertising, which I must say is a superb poster by Steinar, as yet no idea when it will be on the ATARI shelf or, whatsmore, if it will International Karate another of their games has been shown, and seems to have been an initial success – do hope this is not going to be another case of now you see it, now you don't!!!

•If you haven't had enough of snooker, Steve Davies Snooker is available both on cassette and disk.

•ST Software seems to be arriving thick and fast now BRATACCAS, The Interactive Video – is available now – price around £34 00 with a small poster – Roger Dean's superb ILLUSTRATIONS/DRAWINGS were on show at the ATARI Computer Show – both game and illustrations looked very good.

•Other ST games around at the moment are THE PAWN £24 99 and the Microdeal range, MUDPIES, FLIP-SIDE and LANDS OF HAVOC all at £19 99

the software supplies company

THE SOFTWARE SUPPLIES CO is a "one-stop" supplier of specialist computer software, with a customer base of thousands world wide. Since its introduction we have specialised in software for the ATARI ST, and an impressive selection is now available from us at really attractive prices. If you see us exhibiting ST user - or thinking of becoming one (check our hardware prices!) - get on to our mailing list straight away for up-to-date information on what's available for those musicians amongst us. Pianos, instruments include voice and video digitisers, various very sophisticated CAD packages, and expert systems. Latest details will be found in our catalogue which will be sent to active customers only or with orders. Specialist support schemes are available for most of the true business software, from actual installation and commissioning to "hot-line" maintenance please ask.

UK FIRST-CLASS POSTAGE: 75p for all software orders up to £30 and free for prepaid UK mainland orders. All other orders subject to postage at cost with the following minimums £1.50 for orders up to £50, and £2 for orders over £50. For Europe double the rate. Airmail/Eurapid elsewhere at cost.

PAYMENT: Please make cheques/POs payable to THE SOFTWARE SUPPLIES COMPANY or use your Access (Mastercharge) or Visa card. Letters to 70 Box 19 Whitstable, Kent CT5 1TJ or phone your order by 'phone (21-hour, part answerphone). Telex on Gold Hallbox 72:dtb10183. Personal callers by appointment only.

Please note that cheque and credit card payments are not processed until orders are ready for despatch.

Please note that the list of software here consists of both existing and planned releases and can not be complete. Products prefixed with [] are imminent releases and we are pleased to take advance orders for these, or will notify you when they arrive.

PRICES

COMPARE OUR PRICES! Providing we are able to supply it, we'll better anything lower on ST software you see advertised in this or other magazines - and on most items offer you PRICE PROTECTION! credit for a month afterwards too!

In the list here, the first price is our STANDARD discount and you can compare this with the usual price (in brackets). The last is say three mths in the DISCOUNT CLUB price, available to registered club customers on those wishing to join with their initial order (please see conditions below). Some items are not available at special discount, or at standard discount.

HARDWARE

ST computers, mains hardware, and peripherals are available at discount prices from us - please phone for latest prices and availability. Those products are fully supported and guaranteed latest available versions. Local delivery or personal collection can be arranged in place of insured parcels / mail delivery. Special "cash & carry" prices also available in computer systems: please ask for details. Additional free software is supplied with computers at the prices here (see below).

1040STF + SM124 mono + software	£580
1040STF + SM1224 colour + software	£780
SF957M keyboard unit + software	£380
SF351 half megabyte drive	£140
SF314 megabyte drive	£185
SH 20 twenty megabyte hard disk	£800
4(40)B Hobo hard disk [Job]	£750
SM124 12" hi-res mono monitor	£140
SM1224 12" med-res colour monitor	£375
520STM + SF351 + SM124 + software	£690
520STM + SF351 + SH124 + software	£810
520STM + SF314 + SM124 + software	£780
520STM + SF314 + SH1224 + software	£925

520STFM	£ins
Fidelity CM14 colour monitor	£200
Fidelity VM1400 colour monitor/tv	£230
SM1533 med.res.RGB + lead	£285
Philips CM8533 med.res.RGB + lead	£285
Philips monitor (Centronics parallel L)	£20
Mouse mat in	£20
Modem - RS232 cable	£20
Dust covers -disk, h/lrd, mono- each	£3
Monitor/disk lead accept	£30
Printer stand (matches above)	£25
Memorex/3M - ss/dd 3.5" each	£3
Memorex/3M - hd/dd 3.5" each	£5
Plus storage box (lockable, 80 disks)	£10

Delivery is free in the UK for prepaid hardware orders, carriage on all credit orders is charged at cost.

Please enquire about special hardware + software packages where greater savings may be possible. All computers are supplied with the current bundled software, plus extra software as a special surprise from Software Supplies! The latter ranges in value from £30 to £100.

ATARI ST SOFTWARE LIST

- [] A Mind Forever Voyaging - [not yet] Infocom sci-adventure 149 [111] £38
- [] Alternate Reality - The City — £26
- [] Analyst/Writer — £26
- [] Arazok — adventure — 145 [£44] 140
- [] Pedro — full word processor — 15ns
- [] Spindizzy — — £28
- [] Bobfit — full fantasy on which can [online progress] not [but — 115 [£29]
- [] Barnard Term — — 147
- [] BOS BUSINESS SYSTEM — diTails + form
- [] Bureaucrat — clean adventure 132 [135]
- [] Cash Loader — easy to use accounts program 115 [with Analyser 125]
- [] Cash/Sal Accounts — integrated professional accounts — £31 [£39] £309
- [] Chat — — 15ns
- [] Colourpaint — spectacular [and] — mythologiser utilities. — £28
- [] Cosmi 68 — — £30
- [] Crimson Crown — £33 [£39] £33
- [] Cutthroats — £33 [145]: £33
- [] CAD-3D — full 3D [model-building] [Calibr] with [possible animation — £ins
- [] COLR Object Editor — £32 [£35]
- [] Datastar — — £55
- [] Integrate 2 — £30
- [] Deadline — [Infocom adventure — £19 45
- [] DEGAS — design & animate[sprites & graphic art] system. £39 [£39] 135
- [] Delta Patrol — — £33 [£36]
- [] Desk Thing — [Ten temporary] 155 [£40]
- [] DFT [£39 excise into transfer — 12s-95
- [] dual Doctor — [online — £28 [£30]
- [] Disk Help — [simple to use] file repair [+ data recovery utility] — £29 95
- [] Dragon in FORTH Level 1 — £15 [£18]
- [] Dragon Ag FORTH -Level 2 — [£18] £25
- [] Eagle FORTH Assembler — £26 [£25]
- [] Easydraw/d — adventure — 112 [119]
- [] Easyplan — very high quality charting package for [designers/architects] — [£45]
- [] Enchanter — Adventure — £35 [£38] 132 50
- [] Expert Opinion — [expert system for problem solving - 300 rule — £ins
- [] Express Letter Processor — £47 [00 — £25]
- [] Fahrenheit 451 — adventure — 112 [£14]
- [] Font It "451] — 180-dim system, Pascal-like structure, cartridge — £40
- [] Flash — — very sophisticated programmable terminal program — £ins
- [] Flood Stuart Stolen — — 15ns
- [] Fifth Side — version of the popular board game Reversi — [1]4 95
- [] Forbidden Quest. — — £35
- [] Optrex — — £30
- [] FST C Compiler - Lowpiler, assembler [link, editor, shell — £55 [£40]
- [] GST Fractal Generator — [all C and assembler source code — [1]1 95
- [] [BI Kaos Assembler — Motorola compatible, fast [enough]! — £37 [£42]
- [] Mini Batbman Editor — — letter editor for Infocom [1]0
- [] Hint Checklistbox — always at hand [account] system — 145
- [] Hotchas — — [plus] records processing system, Hotel [Travel — £39 [£37] 130
- [] Hitchhikers — — [online Gal] Travels [Pan Sex and Dolfin — £31 [£40]
- [] Halsall[ls... — [all personnel] [4]0 [£43]
- [] Hackwriter — [easy to] use accounts program [115 [with Analyser [£30]
- [] [Analyst] Accounts — [integrated professional accounts — £31 [£39] £309
- [] Chat — — 15ns
- [] Colourpaint — spectacular [and] — mythologiser utilities. — £28
- [] Cosmi 68 — — £30
- [] Crimson Crown — £33 [£39] £33
- [] Cutthroats — £33 [145]: £33
- [] CAD-3D — full 3D [model-building] [Calibr] with [possible animation — £ins
- [] COLR Object Editor — £32 [£35]
- [] Datastar — — £55
- [] Integrate 2 — £30
- [] Deadline — [Infocom adventure — £19 45
- [] DEGAS — design & animate[sprites & graphic art] system. £39 [£39] 135
- [] Delta Patrol — — £33 [£36]
- [] Desk Thing — [Ten temporary] 155 [£40]
- [] DFT [£39 excise into transfer — 12s-95
- [] dual Doctor — [online — £28 [£30]
- [] Disk Help — [simple to use] file repair [+ data recovery utility] — £29 95
- [] Dragon in FORTH Level 1 — £15 [£18]
- [] Dragon Ag FORTH -Level 2 — [£18] £25
- [] Eagle FORTH Assembler — £26 [£25]
- [] Easydraw/d — adventure — 112 [119]
- [] Enchanter — Adventure — £35 [£38] 132 50
- [] Expert Opinion — [expert system for problem solving - 300 rule — £ins
- [] Express Letter Processor — £47 [00 — £25]
- [] Fahrenheit 451 — adventure — 112 [£14]
- [] Font It "451] — 180-dim system, Pascal-like structure, cartridge — £40
- [] Flash — — very sophisticated programmable terminal program — £ins
- [] Flood Stuart Stolen — — 15ns
- [] Fifth Side — version of the popular board game Reversi — [1]4 95
- [] Forbidden Quest. — — £35
- [] [BI Kaos Assembler — Motorola compatible, fast [enough]! — £37 [£42]
- [] Mini Batbman Editor — — letter editor for Infocom [1]0
- [] Hint Checklistbox — always at hand [account] system — 145
- [] Hotchas — — [plus] records processing system, Hotel [Travel — £39 [£37] 130
- [] Hitchhikers — — [online Gal] Travels [Pan Sex and Dolfin — £31 [£40]
- [] Halsall[ls... — [all personnel] [4]0 [£43]
- [] Hackwriter — [easy to] use accounts program [115 [with Analyser [£30]
- [] [Analyst] Accounts — [integrated professional accounts — £31 [£39] £309
- [] Hackwriter — [easy to] use accounts program [115 [with Analyser [£30]
- [] Hacker — Rough [successful game — 112
- [] Henry's [Intelligent] DESIGN — [edit] of FASM/FASM-M — £35 [£31] £00
- [] Hug — Cuthbert [type game — £37 [0
- [] Higgo Planner — [1]4 [00 triplet] [answer] to [chinese questions. — 115 [115]
- [] Higgo Deployment — — £35 [£39]
- [] Higgo [Journal] — — [Chinese processor] [write up to] [handwhy] [concepts — £[the
- [] Hippo-Forth — [Ten temporary] 155 [£40]
- [] Hippo-Graph — [easy to use] file repair [+ data recovery utility] — £29 95
- [] Hippo-Pascal — [simple

DB Master One

from Stoneware
Review by Mike Stringer

Over the past few months quite a number of 'free-bees" have dropped onto the mat from ATARI These include BASIC two Word Processors — 1st Word and ST Writer and a Database — DB Master One

The specifications for this piece of software are very impressive

Maximum field size	320 000 characters
Maximum number of fields	100
Maximum field size	3,000 characters
Maximum record size	3,000 characters

One can fully appreciate ATARI's logic of including a Database program as part of a software package, comparable programs can cost a great deal to purchase if the customer has the need — and it would appear that most of them do I most certainly have a need for one

Included with the disk is a comprehensive 50 page manual It covers all of the principal features of the program, although in a very 'off-beat' manner It takes some time to become accustomed to the author's style I would have preferred to spend a couple of hours following the creation and manipulation of a file, which may be useful to have on hand for some future use, than the method that was chosen — blind faith!

The manual itself is well illustrated, in fact this is the best feature There is an index, but the references back are often vague and incomplete

The program consists of two parts: "MAKE ONE" and "USE ONE" As their titles imply, a file is created by MAKE ONE and in use, USE ONE is the working one.

Make One

On booting up the title screen appears and the rest of the program continues to load when a key or the mouse is pressed. The menu line consists of Desk, File, Edit, Splat and Options.

Desk

This is the menu that is familiar to 'lity programs but with no additional desk accessories

File

New is selected when a new report is to be started.
Save saves the new design to file
Redesign as the name implies, allows the user the option to re-design a report currently on file
Quit returns the operator to the desk top

Edit

Cut Field removes data from a field.
Copy Field used with Paste.
Paste field places the recently cut or copied field to the new position.
Cut Text removes text.
Copy Text used with Paste
Paste Text places text in the new position

Splat

Mailing list	ready made address report
Checkbook	ready made cheque handler
Collection	ready made report for stamp, coin, art, & antique collectors
Date	ready made report for Year, Month, Day
Warthog	ready made fun report, only useful if you are interested in warthogs!

Options

Label Style	this option contains a palette and type styles to design Label Text
Data Style	uses the same palette to create the finished appearance of the data in the field.

There is a great choice available in this option There are six type styles to choose from, at least four colours and 6 sizes The data style inside a field box can differ from the style chosen for the field Label, but styles can even be mixed within a given field Considering that a report can consist of 100 fields, it is very easy to have a different presentation for each one!

Creating a report is not difficult, although some time and experimentation is required before it is presentable While constructing the data box, especially after moving it around, a very annoying feature is evident. The data box defaults to a full line width! This is fine and dandy if that is what is required, but in most cases, it isn t. It reverts to the shape you designed the next time you click on the screen Problem A new field is then created, which has to be removed! This problem should have been removed long ago.

Once a satisfactory report form has been created then it is saved to the file and unless another is to be made, we will proceed to Use One and examine this part

Use One

This program loads in the same manner as the previous one and the file is selected from the ONE menu and loaded. The report form that has been created now appears with the cursor (faint) in the data box of the first field in the report

The menu associated with this program consists of of Desk File, Edit, Report and Record

File

The first line of this menu shows the number of reports that the file contains and the percentage of available memory that they occupy

Switch File	This option quits the current file with a save option to the ONE menu for further selections
Save	this option allows you to backup your file as it is being created without quitting
Quit	returns the operator to the Desk Top

Edit

Cut removes data from the field selected
Copy used with Paste
Paste places the most recent cut or copy text to the field selected.
Clear Field erases the data in the selected field.

Report

Design New allows the user to create and save up to ten different forms for final presentation to the screen, printer or disk.
Edit is an option for modifying the current report form
Print outputs the file to screen, disk, or printer

Record

Add puts a blank form on to the screen.
Find All presents all the current forms to the screen which are paged by the NEXT or control N
Find Some presents one, or a group of selected forms to the screen paged, if necessary, as above
Next pages the next report on file
Previous pages backwards.
Delete erases the report currently on the screen

The cursor is automatically positioned in the data box of the first field of the report. When the data has been entered, pressing return moves the cursor to the next field In this manner the report form is filled. When full, there are two ways to select another blank form With the mouse select RECORD

and ADD or press control A In practice, I find that the latter is more useful. There are a number of such control key options available all of which are clearly noted in the manual, or on the numerous HELP options that are found at the bottom of all the drop-down menus or the HELP key

Another annoying bug relates to the HELP options I make it a habit to lock, from the desk top, all my programs This program will inform you that the locked HELP program has been removed from the disk!! Bad programming has not allowed for locked files – BE WARNED!!

Once a number of reports have been entered, they can be presented in a number of ways The SORT option is quite extensive and, with a little practice it can be quickly mastered. To print the file to the screen printer or disc a Report Form must be designed

This is quite simple Select, with the mouse, REPORT and also DESIGN NEW The screen then splits, with your report in the top portion and the construction screen below If your report form is larger than the screen you will have to use the mouse in the right-hand size column

Clicking on FORM gives you the choice of four form designs. These are COLUMN PAGE RECORD and MAIL LIST The latter can only be used if you used the MAIL LIST report form from the SPLAT menu in the MAKE ONE program and prints to a PRINTER not to the screen REPORT too can only be shown on a printer This leaves just COLUMN and PAGE that can be shown on the screen, printer and disk This is certainly adequate for most end users

The presentation of Column is as its name suggests a form with the data arranged in columns The data in the fields is not fixed If the arrangement exceeds one page width data is removed from each field, from the right, until it does fit.

Selecting FIELDS with the mouse the manner in which the fields appear on the document can be arranged Another option can be introduced at this time, FIND Selecting this you can specify a group of records to be printed SORT allows up to three levels of sorting, with a further option of 0 to 9, A to Z or 9 to 0 Z to A

Additionally subtotal and new page options are available If you are satisfied with the presentation, select REPORT and PRINT This produces an option page where you NAME the report form just created After entering the name, the screen changes once more to a directory plus options You may then select SCREEN PRINTER or DISC The first two are self-explanatory but the third – DISC is quite novel Selecting this option produces a DOC file This can be introduced into a word processor program or printed to the screen from the Desk Top PAGE fixes the data in each field, consequently if the selection exceeds a page width, an error message appears Each report is allocated a separate page

RECORD sets out on the printer an exact copy of the report form as it was created, complete with all the various data and label print style and sizes

MAILING LIST prints out to standard one-across, six line sprocket fed labels. The form needs some small modifications to make it suitable for use in this country – it is based on the United States of America's addressing system!

DB Master One in Use

As I mentioned earlier I have a need for a good powerful data base One of my pastimes is Philately, specializing in Postal Mechanisation To the uninitiated this boils down to a study of the pretty rows of blue dots seen on envelopes I have a considerable number in my collection which lend themselves to database manipulation After some experimentation, I started to fill in report forms, making numerous stops to back-up my work I am very glad that I did After entering some 1500 records I printed to the screen, all of these records Only 5 appeared. I checked, rechecked, checked again, but still only 5!! I even tried to design a new form. This time the program bombed out! With the new TOS, it literally bombs out No fewer than four bombs appeared, (they used to be mushroom clouds!) and I was dumped unceremoniously on the desk top!

Experience has taught me that when these little beasties appear, there are problems BIG problems! The higher the number of bombs – the greater the problem This is a very nasty little bug to discover. Confidence in the program is shattered and I now use it with great caution BE WARNED!

I mentioned the back-ups I am glad that I not only saved, using control–S, but also made physical backups to other disks I have been bitten before!! I only had to re-enter a couple of hundred – a few hours work, and all was honky dory again

This goes to prove that frequent saving and backing up of your files is a good habit to develop Another nasty bug has cropped up quite frequently, again with lots of little bombs when editing a report form for printing It usually appears during a re–FIND and/or a re–SORT BE WARNED AGAIN!

I was very impressed with the speed in which a sort was performed, especially when printing a form to the screen On one file, consisting of 1600 records, 70% full, this was on average only 1 6 seconds. Now by any standards that is fast. Very fast, compared with that which is available on the old 800 or, chuckle chuckle the couple of minutes it takes a BBC Micro to sort three records from 130!! Unfortunately there is a price to pay for this speed. Hassle To print a document to the screen requires no fewer than 12 moves if one checks the sort ,or introduces a different sort.

SELECTION	PROMPT
Report	Move to Edit
Edit	Click
Choose Report	Click
To Edit	Click
Find	Click
Equals	Click
	Type in Heading
Report	Move to Print
Print	Click
Replace Old	Click
Choose Report	Click
Send to Screen	Click

All this paraphernalia increases the speed from 1 6 secs, to 24 secs on average

The other presentation to the screen is FIND This takes 14 seconds, on average The procedure is much less of a hassle – only four operations are required this time

SELECTION	PROMPT
Choose Field	Click
Equals	Click
	Type in Heading
Find	Click

The actual record card is now on the screen and if a group of records were requested they are stacked beneath this card Even with all this hassle, the speed of presentation is the reason why I shall stick to this program for a while

Another niggle but not serious, is the diabolical grammar used in the warning screens. If this program had been written using a RSC file I would have modified the text long ago But it isn't so I can t! The screen presentation is very poor, with a psuedo-window which is uncontrollable You cannot have more than one working file up at a time which would have allowed sections to be transferred from one file to another more easily

Having spent quite a lot of time outlining the program I think it is time for an appraisal On the plus side, I liked the speed, the manner in which documents were printed on the screen (approximately 60 lines X 120 characters !!), the options available and the capacity The dislikes to summarise, are the general presentation, the manual, the bugs, the screen presentation and a general impression that the program has been bodged from another computer to work, in a fashion, on the ST

The gesture of ATARI to include a database in the package is to be commended, but their choice of product leaves much to be desired. I suppose I shouldn't complain too much, it was free after all, and is quite useful with care On a rating of 10, DB Master One gets 4

ST SOFTWARE CLUB

Silica Shop are the UK's No1 Atari Specialists. We have been dedicated to the Atari market for the past 7 years and have the widest range of Atari product in the UK. We aim to stock EVERY Atari compatible product on the market and offer a FREE information service to Atari owners. Through our ST software club we will keep ST owners up to date with the range of ST product which is available as well as details of any special offers. The list below is of the product which we currently have in stock for the 520ST as well as titles which we expect to be released in the near future. If you would like to join our software club and receive a full copy of our price list with release details and further ST bulletins, please complete the coupon below. This service is completely FREE OF CHARGE!

ACADEMY
Typing Tutor
ACTIVISION
Borrowed Time
Hacker
Mindshadow
Music Studio
ADVENTURE INTNL
Gremlins
ANTIC
COLR Object Editor
Disk Doctor
Maps & Legends
Murray & Me
Mon P Me
ARTWORX
Bridge
Golf
ATARI CORPORATION
3D Inferno
Basic Compiler
Breakout
Images
Joust
Star Raiders
ST Cobol
AZTEC SOFTWARE
Speakeasy (Speech Synth)
BATTERIES INCLUDED
Degas
Homepak
BAYVIEW SOFTWARE
Word for Word
BECKEMEYER DEV TOOLS
M one C Shell
BLUE CHIP SOFTWARE
Sensn
Millionaire
Tycoon
BOS
BOS Autoclerk
BOS Autoln Pes
BOS Finnell
BOS Fixed Assets
BOS Inventory Control
BOS Invoicing
BOS Job Costing
BOS Microcodel Compiler
BOS Negotiator
BOS Nominal Ledger
BOS Office Automation Pack
BOS Payroll
BOS Planner
BOS Purchase Ledger
BOS Sales Ledger
BOS Sales Order Processing
BOS Screme
BOS Writer
BOS/S Operating System
B I M SOFT
BBC Basic Interpreter
CASHLINK
ST Accounts
Helmer
CHELTEX SYSTEMS
Small Business Accounts

CHIPSOFT
ST Accounts
CROSSBOW MUSIC
Music Writer
DATABENCH
Optimum
DATACQUE SYSTEMS INT
Project Leader
DELTRON
Home Planetarium
DIGITAL RESEARCH
Calculator
GEM Draw
GEM Paint
GEM Write
ST Basic
ST Logo
DRAGON GROUP
4 x Forth - Level 1
4 x Forth - Level 2
Forth Accelerator
ST Colouring Book
ELECTRONIC ARTS
Financial Cook Book
Marble Madness
ENGLISH SOFTWARE
ST Electrogloss
EXECON SOFTWARE
Taurus Accounts
Taurus Invoicing
Taurus Printers Est/Quote
Taurus Stock Control
EXTENDED SOFTWARE CO
Disk Library
FTL/OASIS SOFTWARE
Sundog - Frozen Legacy
FYLER MICROPROGRAMMING
Administrator
Dalestats 2
Mailmate
Mailograph
Personal Assistant
Spellmate
Spreadmate
GST SYSTEMS
GST ASM
GST C
GST Edit
HABA
Checkbook Minder
Haba Dials
Haba Osc
Haba Hippo C
Haba Letters (Business)
Haba Merge
Haba Shutdown
Haba Wills
Haba Writer
HIPPO
Hippo Accounts
Hippo Art I
Hippo Backgammon
Hippo Computer Almanac
Hippo Concept
Hippo Disk Utilities
Hippo Forth
Hippo Jokes & Quotes

HIPPO cont
Hippo Pixel
Hippo RAM Disk
Hippo Simple
Hippo Spell
Hippo Word
HISOFT
C-Compiler
Devpac ST
Pascal ST
INFOCOM
A Mind Forever Voyaging
Hollyhod
Cutthroats
Deadline
Enchanter
Hitchhikers Guide Galaxy
Infidel
Planetfall
Seastalker
Sorcerer
Spell Breaker
Starcross
Suspect
Suspended
Wishbringer
Witness
Zork I II III
INTELLIGENT SOFTWARE
3 D Chess
ISLAND LOGIC
Music System (Thc)
Polyscan
KUMA
K-Data
K-Ram
K-Seka
K-Spread
K-Word
LASERSOFT
LaserPass
LLAMASOFT
Colourspace
MARK OF THE UNICORN
Final Word (WP Processor)
Flex (Cubert type game)
Mince Editor
PC/Intercom - VT100 Emul
MARK WILLIAMS CO
C Compiler
Lats Wi le
MEGAMAX
C Compiler
METACOMCO
Lattice C Compiler
Macro Assembler
Pascal Compiler
MICHTRON
Calendar
Easy Pesors
Flight Simulator
Gauntlet
IBM ASCII File Transfer
Major Motion
Mighty Mail
MiTerm (Comms Program)
Transfer IBM-Atari

MICRODEAL
Disk Help
Flip Side (Reversi Game)
Lands of Havoc
MiCopy
M/Disk (RAM Disk)
Mishtron Utilities
Mud Pies
Soft-Spool
Time Bandit
Twin Pack (M/Disk & Sft Spl)
MICRO-ED INC
Home Accountant
MICROPROSE
Silent Service
MICROPROCESSOR ENG.
Forth 3dbit
MIGRAPH INC
Draw Write
Easy Draw
MIRACLE TECHNOLOGY
ASCII/Fredo Comms
MIRAGE
Express (Lattel Processor)
Holmes & Duckworth Forth
Holmes & D/with Tool Box
MIRRORSOFT
Pixel Sticol Editor
MONARCH DEVELOPMENT
Photon - Shape/Icon Editor
MOSAIC
Integrated 7
OMNITREND
Universe II
OSS
Personal Disk II
Personal Pascal
Personal Prolog
OTHER VALLEY SOFTWARE
Delta Paint
OXXI
Dll One
Word Wiz
PAPER LOMIC
ST Toolkit (PAL)
PARADOX SOFTWARE
Desk Diary
Mean Mouse
PENDUIN
O- mean Crown
Hyphus
Penstate
Transylvania
PHILON
Basic-C-Compiler
Basic-M Compiler
C Compiler
Gamma
Fortran
Henrys Fundamental Basic
Pascal Compiler
RPG
PLANNER
Project Manager

PRECISION SOFTWARE
Superbase
Superscript
PROGRESSIVE COMP APP
Fig-gal
Graphic Artist
PROSPERO SOFTWARE
Fasck (ISO)
Pro Fortran 77
PRYORITY SOFTWARE
Forbidden Quest
Galaway
PSYCHOSIS
Statecoes
QUEST INTERNATIONAL
Cashtrader
Cashl aidi Analyzer Module
Nominal/General Ledger
Purchase/Creditors Ledger
Sales/Debtors Ledger
Sales Invoicing
Stock Control Ledger
QUICKVIEW SYSTEMS
Customers
RAINBIRD SOFTWARE
Pawn (The)
Starglider
REGENT SOFTWARE
Regent Word
Spell Inc Checker
RISING STAR INDUSTRIES
Business Graphics
CAD/CAM System
Muse
Scheduler
Spreadsheet
Vallas
Val Draw
Val Mail
Val Paint
SCARBOROUGH SYSTEMS
Master Type
SIERRA ON-LINE
Black Cauldron
Kings Quest 1 & 2
Ulims 2
Wrath the Posh
SM SOFTWARE
Filemanager
Spelling Checker
Spreadsheet
Word Processor
SOFTEK/SOFTECHNICS
Fireaus
Rhythm
Timelink
SOFTLOGIK CORP
Blacl o Calanpu
Gamma
Logichain
SOFTWARE FUNCH
Atarinet
Atariword
Mirlanet
Presler Emulation
Print Spooler
Punchpak

SOFTWORKS
ASCII/Prestel Comms
SPINNAKER SOFTWARE
Homework Helper (Maths)
Homework Helper (Words)
SST SYSTEMS
Chat Communications Pack
STONEWARE
DB Master II
SUBLOGIC
Flight Simulator II
Jet
SUPPLEMENTAL
Two Key Accounting
SYSTEMATICS
General Ledger
Invoice
Payroll
Purchase Ledger
Sales Ledger
Stock Control
TALENT
Flexfile
Lost Kingdom Zkul/West
RAM Disk + Spooler
TDI
AnPlus Document Proc
Modula 2
P System
TELARIUM
Amaurot
Dragon World
Fahrenheit 451
Kung Fu
Nine Princes of Amber
Perry Mason ManPsrin Sim
TK COMPUTER PRODUCTS
Monday Morning Manager
TOWNGATE SOFTWARE
Cash Manager (BOS)
UNISON
Art Gallery
Print Master
VIP
VIP Professional
WHITENDALE
WASP Accountants Time Recing
WASP Festiana
WASP Incomplete Records
WASP Insurance Brokers
WASP Integrated Legal Accting
WASP Legal Accounting
WASP Printers Costing
WASP Printers Estimating
WASP Solicitors
WINDHAM CLASSICS
Treasure Island
Wizard of Oz
WORD PROCESSOR
Cim Word
XLENT SOFTWARE
Megafont
Rubber Stamp
Typesetter ST

FREE MEMBERSHIP

JOIN NOW!

SILICA SHOP
THE UK's No1 ATARI SPECIALISTS
SILICA SHOP LTD, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX
Tel: 01-309 1111

FREE PRICE LIST & BULLETIN

To: Silica Shop Ltd, Dept ATCOC 0686, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX
Please enrol me in the FREE Silica ST Software Club and send me your price lists & bulletins as they become available

Mr/Mrs/Ms *Initials* *Surname*

Address

Postcode

I am interested in the Atari ST ☐
I already own an Atari ST ☐

Time Bandit

Reviewed by Michael Stringer

Microdeal's Time Bandit, by Bill Dunlevy and Harry Lafnear, should be available from your dealer by the time you have received this edition of MONITOR.

It is one of the 'New Generation' of computer games combining the traditional arcade 'blast everything that moves', together with, and at times, the essence of text adventure games, requiring the player to ask/answer questions via the keyboard. This, of course tests your wit, logic and riddle solving abilities

There are over 200 screens and levels available, only some of which are text screens, the majority are arcade with the appearance of some familiar 'flavours'! You will understand this observation when you get the opportunity to have-a-go The main objective is to accumulate treasure, avoiding the white transporter disks (nasties have the habit of making unwelcomed appearances from these devices) and also avoid passing over the PORTALS, danger lurks therein

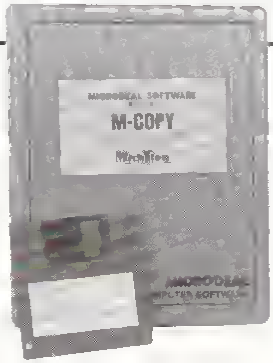
The action is fast and furious, as one might expect, it is also quite colourful The sound routines aren't that great, but who pays much attention to that anyway? The death scenes are quite novel and the graphics are much better than 'Lands of Havoc' There is also a dual-player mode so that two people can play simultaneously, each has his own window This is not for me though, I want this one to myself!

The price is £29 95, still expensive as most new computer software is these days If the pricing was around £20 and also catered for the owners of the high resolution monochrome systems (it is *colour only*), I feel that it would reach a wider public Nevertheless, just about the best game available so far for the ST

Three from Microdeal

Reviewed by Michael Stringer

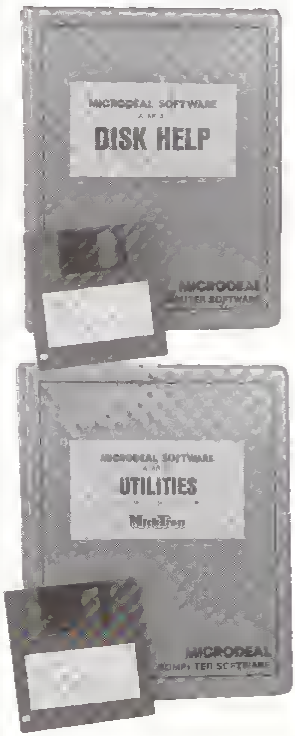
Recently, a batch of utilities from MICRODEAL arrived at the office for review All three were decently packaged, but the manuals were disappointing M-COPY will be a very useful tool for clubs and small, professional outlets, but it is doubtful if it will be of any service to the average ST user M-COPY as the title suggests, is a utility for producing multiple copies of a disk. Provided it has no protection It reads into memory all of the data, skipping unused sectors Now that TOS is resident in ROM on all new machines and available, at a small cost, to those owners with TOS on disk, M-COPY will read one side of the standard disk in a single pass A blank disk is then inserted, formatting and copying are automatic Multiple copies only require the insertion of new disks M-COPY will stop if the program being copied has any bad sectors, or any form of commercial protection, hence the previous com-



ment. In my opinion it is an expensive, at £49 95, program which will have very little use

UTILITIES has better documentation and considering that a sound knowledge of disk architecture is required to utilise all the facilities of the program, it provides some basic information on disk structure. There are some useful facilities in the program, but in the copy received, there were quite a number of bugs. It is likely that this copy is a pre-release version, and that the bugs will be removed in the final version. One of the useful features available is the ability to restore deleted files, either in part, or whole That is, of course, if additional files have not been written to the disk Even so, some of the files may be retrievable This program uses GEM and the numerous functions are obtained through drop-down menus using the mouse Some of these functions are the ability to format individual disk tracks and there is a 10 sector formatting utility which is claimed to add 80K of disk storage to a disk. Changing file names, volume names, attributes or any individual bytes on disk are other claims At £39 95 it is not cheap, but if the bugs are removed it will be a useful program to own These two programs originate from Michtron and the third, DISC HELP, from MICRODEAL.

DISC HELP is a more basic and simple program than UTILITIES, written without using any part of the GEM environment. It is run from a menu and offers some of the features of the previous program. It is a program designed for the non-technical user, but again, the documentation is very poor The price, £29 95, is expensive for such a basic utility



I received from Mike's Computer Store, Westcliff, some free Desk Top accessories the other day. These are just a few of the many that are beginning to appear and will be available to his customers who send him a disk. The first is a simple analogue clock, written within GEM with full sizing and movement possible The clock reads the instruction from the Control Panel, updating it each minute Novel and useful

The second is quite useful, too. It is a thirty year calender, again written within GEM. The line up (down) arrows control the month and the page flip controls the year The third accessory is also very useful, a basic calculator It has an eight character display, the usual +, −, * and / with CE and CLR functions. It does not have any sizing facility none is required. Another accessory is entitled TINY TOOL! No ribald comments from the back, please!! This useful program is written by A Birtz. It is another mouse driven disk utility, powerful with a useful print dump facility The final one I shall mention should be kept on a chain. It is called 'CRABS' When a disk is booted up, the crabs are released! There are eight of the little monsters wandering around the screen gobbling up everything in sight Not very practical, but 10 out of 10 for amusement value The desk top is quite practical and interesting If you can spare about 40K of RAM then I recommend them thoroughly Congratulations to their respective authors No doubt there will be others along to replace them in due course, but in the meantime they will serve admirably My thanks to Mike Wilding of Mike's Computer Store and to MICRODEAL for the material they kindly sent.

MENU+

Reviewed by Michael Stringer

A new programming utility from METACOMCO arrived recently and I have had the opportunity to spend a short time to get accustomed to it. The title "MENU +" is quite modest. This is a powerful applications program designed to speed up the development of a program from the editing stage, through the compilation, linking (where necessary), to the running of the program

I have not had sufficient time with this program to try it out on other compilers, but it works admirably with Metacomco's compilers The program runs under GEM, and the user has full control, through a very efficient designing utility built in, of all the various tools and sections that are made available.

It hes three specific menus available, and more will no doubt become available later, for Assembler, PASCAL and C I found it very easy to use without any re-designing of any tools or options, but no doubt these would not suit all applications It was also much quicker If you are familiar with the tedious regime demanded by some compliers to arrive at a working program, then one feature of this program will be most useful you can set a Batch File to COMPILE, LINK and produce a PRG in one request. In doing so, it provides a homogeneous programming environment.

Included on the disk is a utility program which allows copying of the file to the console, printer, deletion, renaming and copying

The manual is also included on the disc as a .DOC and is quite comprehensive demonstrating quite clearly some of the many permutations that are possible MENU + will be supplied with all future releases by METACOMCO I gather, and to existing owners, a modest charge is being made

I think that this will be a most welcomed program for all serious users of C, PASCAL and ASSEMBLER compilers where speed, accuracy and convenience is demanded, and at only £19 95 it's well worth the money

OPENING OUT

AN INTRODUCTION TO THE USE OF FILES Part 3

By Ron Levy

Part three of OPENING OUT is devoted to explaining the Atari's disk system, and how to take advantage of some of its special open modes to create and use random access files

In it's basic concept, a disk drive has a resemblence to cassette storage, in that they both store data in a sequence of 128 byte blocks This, however, is where the similarity ends, for there are two very important differences between tape and disk storage. Not only are disks very much faster than tape, but they also allow *random access*.

What is Random Access?

Consider a file on a cassette tape. Suppose that the computer is asked to read a record at the other end of the file, it would have to read through lots of unnecessary data before it could reach the part it wanted – it would certainly not have a very good response time With a disk, however the read head can move directly to the part of the disk drive it wants within a fraction of a second But how does the computer know where to go on the disk drive? Well the key to this is *organisation*.

The Disk's Structure

Let's use the single density (as in the 810 drive) as our working example The disk holds the data in clusters of 128 bytes, just as in a cassette system, and we call these SECTORS Now these sectors are arranged in circles around the disk tracks There are 18 sectors in each track, and there are 40 tracks on the diskette, thus giving a grand total of 40 x 18 = 720 sectors, or 720 x 128 = 92 160 bytes of storage space! Wait a minute though, I hear you say, why is it then that when a disk is formatted, it only has 707 sectors free to use?

Since the floppy disk is a random access device it is also able to store more than one file on each disk, but somehow it needs to keep a record of the files it holds, where they are on the disk, and how long they each are It does this by reserving an area near the middle of the disk called the DIRECTORY

The directory occupies 8 sectors, and each directory holds information on up to 8 files giving a maximum possible number of 64 files One sector is also used to hold a bit-map of which sectors are in use, and which ones are free This is called the VOLUME TABLE OF CONTENTS, or VTOC for short. Three or four sectors are also used at the start of the disk to keep other information, but this need not concern you yet.

This, then, is how you 'lose' sectors on your newly formatted disk. The story does not end there, however Remember that I said each sector holds 128 bytes, well when you save a program or file to disk you will only be using 125 bytes out of each sector to store your data! The 'lost' three bytes are used by DOS (Disk Operating System) to point to the next sector in the same file This means that a file does not have to exist over a consecutive block of sectors, but instead is able to be split across parts of the disk, as free space is found. There is also a value in these three bytes which tell's the computer which file the sector belongs to

Opening Modes

You will, I hope, recall from last time the format of the OPEN command Just to remind you, here is an example

```
OPEN #1,8,0,"D:TEST.DAT"
```

This will open the file TEST.DAT on drive 1 for OUTPUT We know that it is for output because an 8 is placed in the mode/data direction byte If it were a 4 then it would be READ mode Take a look at Table 1, and you will see that there are in fact several more modes available I will now explain the meaning and use of each

a) Mode 4 - Input

In this mode the computer will look on the disk for the specified file, and if it does not exist will return error 170 – file not found, it will not create a new file If a file is found, any read operations you perform will of course begin at the start of the file You will not be allowed to write to the file (using PRINT #1 or PUT #1), any attempt to do so will result in error 135

b) Mode 8 - Output

With mode 8 you can create and write to a file If a file, of the same name already exists on the disk, this file will first be deleted (just as if you use the DELETE FILE command on the DOS menu!), before the new file is created This is a common error for beginners, resulting in the loss of data and often much heartache! If the original file is locked, however it will not be deleted but instead error 167 will be returned If the file is opened and created successfully, it will have zero length until you start writing records to it.

c) Mode 9 - Output Append

Mode 9 is similar to mode 8 in that

Code	Function	Meaning
4	Input Only	File pointer is positioned at the start of the file
6	Disk Directory Read	
8	Output Only	File pointer is positioned at the start of the file, creates a new file, any existing file is deleted
9	Output Only, Append	File pointer is positioned at the end of the file, the file *must* already exist.
12	Input and Output	Positions file pointer at the start of the file, data can be read or written to in this mode The file must already exist, and it cannot be extended.

Table 1

you will only be allowed to output data to the file, but the difference is that the file MUST already exist, and that as you PRINT or PUT data out it will go onto the end of this file In fact, if the file does not exist then error 170 will be returned This mode is the one you must use if you want to increase the size of an existing file without deleting it or altering the data that is already there

d) Mode 12 - Input/Output

This mode allows you to both read *and* write to a file. Once again, the file specified must already exist on the disk. Once the file has been opened your program can either read data from the file, or write over (to modify) data on the file When the file is opened the pointer is positioned at the beginning of the file There is no restriction upon the sequence of 'reads' or writes' to your file, so great care is needed to maintain the integrity of your data, but once mastered, mode 12 is the most useful available to you for working with random access files All that is required is some means of moving the file pointer around at random – more about this later!

e) Mode 6 - Disk Directory Read

This mode is provided to enable you to read the contents of a disk's directory in your program. (Yes, it can be done from BASIC!) Once OPENed, the directory can be read just like a text file, simply by using the INPUT# command The strings received can be printed to the screen, and their format is identical to the directory listing obtained using the DOS menu If you are using a DOS produced by Atari you probably find it infuriating to have to save your program, insert a DOS diskette, and exit to DOS simply to look at the contents of a diskette Well, one way of getting around this when developing a program is to have a directory read/display subroutine somewhere near the end of your program. Take a look at Program 1, this is a routine which you can call by just typing GOSUB 32700 Notice that I have used the GET command rather than INPUT; I have done this simply to avoid having to DIMension a string! If you LIST this subroutine to a disk file, you can ENTER it onto any existing program you may have

File Pointers

The computer knows where to read and write to next on a disk file by maintaining an invisible file pointer Since, as I have explained, a disk is composed of sectors numbered 1 to 720, and bytes in each sector numbered 0 to 124, the computer has to use two numbers to describe this pointer. You may remember, from Part 3, that the

```
OS 0 REM *** PROGRAM 1 ****
OF 1 REM DISK DIRECTOAY READ DEMO
NH 2 REM
JL 10 GOSUB 32700:STOP
AZ 20 REM
PU 32700 REM *** DISK DIRECTOAY ***
JT 32710 OPEN #5,6,0,"D:*.*"
OK 32720 TRAP 32750
BN 32730 GET #5,XX:PRINT CHR$(XX);
EI 32740 GOTO 32730
FA 32750 CLOSE #5:RETURN
```

computer used a buffer for each file that had been opened Well, in the case of disk files each sector represents one 'buffer load' in size. In fact, the sector pointed to by the file's 'current sector' pointer will actually be in memory, *in the buffer* for that file

Taking Note

I have explained that the computer keeps track of where it is in a disk file with a two number pointer, and it is probably quite obvious that in order to achieve random access across a file we must be able to look at these numbers, and force them to different values The question is, how can we do this from a BASIC program? The answer lies with

```
PL 0 REM *** PROGAAM 2 ****
AC 10 DIM A$(50)
WJ 20 OPEN #1,8,0,"D:T1"
KT 30 AEAD A$
BG 40 IF A$="END" THEM 32767
KO 50 NOTE #1,X,Y
YQ 60 PRINT "S=";X;" B=";Y;
XO 70 PRINT CHR$(127);A$
RS 80 PAIRT #1;A$
SI 90 GOTO 30
OU 300 DATA THIS IS A SIMPLE DEMO
ME 310 DATA DEMONSTRATING
AF 320 DATA HOR THE NOTE COMMAND
IG 330 DATA OPERATES BY
VF 340 DATA PAINTING THESE DATA
MP 350 DATA LINES
LD 360 DATA TO A DISK FILE AND
HC 370 DATA PAINTING THE RESULTS
KH 380 DATA OF THE NOTE CORRARD
MC 390 DATA SO THAT YOU CAM SEE
EA 400 DATA HON THE SECTOR
RB 410 DATA AND BYTE VALUES
AA 420 DATA CHARSE.
ER 430 DATA YOU CAR OF COURSE
DH 440 DATA ADD AS MUCH TEXT
HZ 450 DATA TO THESE DATA LINES
BY 460 DATA AS YOU WISH BUT
QF 470 DATA THE LAST DATA
EY 480 DATA LINE MUST BE ERO
IU 999 DATA ERO
AD 32767 END
```

two very valuable, commands, namely NOTE and POINT

First, let's take a look at the NOTE command This command is used to obtain the value of the file pointer, so it will need two variables. The format is as follows

NOTE #1,X,Y

In this example the current sector number of the file open on channel 1 is placed into tha variable X, and the next byte to be used in this sector (which is in memory in a buffer) is placed into the variable Y

Type in and run Program 2 Don't forget to insert a disk before you type RUN though

Program 2 OPEN's the file "D:T1." in mode 8, so it will create a new file of that name (if you already have a file of that name on the disk then it will be deleted, remember) It then proceeds to print a series of strings to the file but before each string is printed to the file the program looks to see the sector number and byte number that will be used to store it. These, along with the string itself, are printed onto your screen Write these down on a piece of paper, because you will need them for the next program

Something which you ought to notice is the sequence of these numbers You should see that the byte position increases each time by the length of the string printed plus one This extra byte is the RETURN character code 155, (this is how the computer knows the end of each string when you INPUT them from the file) The byte number will re-start from one when it exceeds 125, and you should notice the sector number also change at this point. If the next sector on the disk is free, then it will increase by one, but if the file straddles the directory sectors, or the sectors of another file, then it will of course change by a larger value.

Reading Our File

We have seen how to create a file and have noted the positions of strings on the disk, so let's take a look now at how to read back our data in any sequence Type NEW and then enter Program 3

```
OE 0 AEM *** PROGRAM 3 ****
PS 1 REM READ THE FILE RANDOMLY
AC 10 DIM A$(50)
UN 20 OPEN #1,4,0,"D:T1"
PO 30 PRINT "SECTOA ";:INPUT X
RF 40 PRINT "BYTE   ";:INPUT Y
IB 50 POINT #1,X,Y
RN 60 INPUT #1,A$:PAINT A$
XH 70 PAINT
SH 80 GOTO 30
```

Program 3 will ask you for the sector and byte co-ordinates for the string you want and you should enter the numbers given by Program 2. You can enter these pairs of numbers in any sequence you like. The computer will move its file pointer to the position you give, and the subsequent INPUT command will read the string at that location. This is true random access!

Notice that the computer only has to load a sector from disk when the sector number changes as well as the byte number. Try entering a valid sector number, but a byte number that is three more than the corresponding byte value. This time you lose the first three characters of your string. This is allowed because when you use the POINT command the computer does not care that you are not at the beginning of a string, it only checks that the sector is really part of the file.

Try entering a sector number that is not part of the file (i.e. not printed by Program 2). Notice that this time, after it has loaded the sector, the computer stops with error 164. A glance at your DOS manual will show that this means FILE NUMBER MISMATCH. Remember how the last three bytes in each sector are used by DOS for file information? Well, one of the functions performed by these is to store the file number that the sector belongs to (a number from 0 to 63, corresponding to the directory entry position), and error 164 is returned if this number does not correspond to the directory entry number. This error also occurs if a file's *next* sector pointer gets corrupted somehow, or if one file has overwritten part of another file's disk area. This can happen if disks are swapped halfway through saving a program, or while a file is still OPEN for output.

Updating our Data

The next question which is on your mind must surely be how can we randomly change the contents of our file? Well, it is, in essence, very simple indeed. In Program 3, change the OPEN mode from 4 (read only) to 12 (update mode). Now, after you have POINTed to the correct string, the command PRINT #1,X$ can be used to print the contents of X$ over the existing string on disk.

This method, however, presents two problems. Firstly, we do not want to have to write down, and then enter, lots of NOTE and POINT values, and secondly, suppose that we want to replace a string in our disk file with a longer string? If we tried to do this we would over-write the beginning of the next string! The answer, however, lies in *good data management*. We must decide the maximum size to allow for each string and fix all the files string sizes

to this value.

Let's suppose that we want to create a file to store up to 50 names, and allow each name to be up to 30 characters long. First we must write a program to create our file. Type in Program 4 and RUN it.

Program 4 will in fact create two files. The first, NAMES.DAT, contains 50 blank strings, each of 30 characters long. The second, NAMES.CTL, is created by printing all the sector and byte numbers used in the main data file (NAMES.DAT).

Now you are ready to enter and RUN Program 5. This is the random access/update program. Using this you can look at and modify any of the records in the file NAMES.DAT. The program works by holding all the sector and byte co-ordinates in a two dimensional array (or matrix), these being loaded from the file NAMES.CTL at the start of the program.

Remember that I said with mode 12 you cannot extend the size of a file? Well

```
QX 0 REM *** PROGRAM 4 ****
XV 1 REM CREATES A NAMES FILE
CO 2 REM AND ITS POINTER INDEX.
RI 3 REM
ZO 10 DIM R$(30),P(50,1)
EU 20 FOR A=1 TO 30
BB 30 N$(A)=" "
FY 40 NEXT A
ZQ 50 OPEN #1,8,0,"D:NAMES.DAT"
BC 60 FOR A=1 TO 50
KO 70 NOTE #1,X,Y
ET 75 P(A,0)=X:P(A,1)=Y
YF 80 PRINT #1;R$
GO 90 NEXT N
KX 100 CLOSE #1
VQ 110 OPEN #1,8,0,"D:NAMES.CTL"
QM 120 FOR A=1 TO 50
KR 130 PRINT #1;P(A,0)
LL 140 PRINT #1;P(A,1)
CX 150 NEXT A
LJ 160 CLOSE #1
```

```
RQ 0 REM *** PROGRAM 5 ****
JX 1 REM ALLOWS RANDOM READ
OR 2 REM AND UPDATE OF OUR
HN 3 REM NAMES FILE.
GT 10 DIM N$(50),P(50,1),X$(1)
HK 20 OPEN #1,4,0,"D:NAMES.CTL"
FX 30 FOR N=1 TO 50
BL 40 INPUT #1,X,Y
EH 50 P(A,0)=X:P(A,1)=Y
GH 60 NEXT A
LR 70 CLOSE #1
GQ 80 OPEN #1,12,0,"D:NAMES.DAT"
OX 90 PRINT :PRINT "TYPE E TO END."
VY 100 PRINT "WHICH ENTRY (1-50) ";
AJ 110 INPUT N$:IF N$="E" THEN 32767
SG 120 N=VAL(N$)
UA 130 IF N<1 OR N>50 THEN 100
CP 140 POINT #1,P(N,0),P(N,1)
ZT 150 INPUT #1,R$
OI 160 PRINT R$;" = ";R$
BR 170 PRINT ,"ENTER N TO MODIFY...";
JO 180 INPUT X$
EE 190 IF X$<>"N" THEN 90
TO 200 PRINT
JV 210 PRINT "ENTER THE AMENDED NAME"
SD 220 PRINT N$;" = ";:INPUT R$
CO 230 POINT #1,P(N,0),P(N,1)
FL 240 PRINT #1;N$
GP 250 GOTO 90
KZ 32767 CLOSE #1:END
```

if, for example, you needed to add more records to the file at some later date you would have to OPEN the file in mode 9 (output append), print more of the 30 byte strings to it, and then also extend the NAMES.CTL file in the same way.

In Conclusion

This name storage program is of course really just a rough skeletal example, (although it works well!) and in a future issue I will give you a more elaborate database to experiment with. In the meantime though, why not try experimenting with different file layouts – see if you can create your own database system!

Until next time.

USER GROUP SOFTWARE

Software Librarian - Roy Smith

Due to demand from members there are now two ways to get programs from the library The original scheme of exchanging 3 for 1 will still apply but now with an added bonus So the library rules have been extended to enable those members who cannot write their own programs to gain access, and those that can to have a possibility of some reward for their efforts The extended library rules are as follows

3 FOR 1 EXCHANGE

1 Every program you donate to the library entitles you to three programs in return
2 The program you donate must be your original and not copied
3 Your donated program must be submitted on a cassette or a disk programs in the form of print outs will not be processed
4 If your program requires any special instructions they should be added in the form of REM statements within the program (or you may present them as instructions when the program is actually run)
5 BONUS Every program donated per quarter (between issues of the newsletter) will be eligible to be judged STAR PROGRAM for that quarter This carries a prize of £10 which will be paid to the author from the club funds The programs will be judged by the Editorial Team and their decision will be final. The Editorial Team are not eligible for the prize
6 The '3 FOR 1 exchange is only open to club members

DONATION SCHEME

1 Every club member will be entitled to ask for up to 3 programs per quarter from the library by donating to the club funds
2 If a member does not take his/her entitlement for a particular quarter, it cannot be carried forward to the next quarter
3 A member can have more than one quarter s entitlement at one time (up to a maximum of 12 programs (1 year)), but then will be unable to ask for more until his/her credit quarters have been used. Note that odd numbers of programs will be counted in quarters i e if a member asks for 5 programs the first 3 will be that quarter's entitlement the next 2 will be the second quarter's entitlement and he/she will have to wait until the third quarter before he/she is entitled to any more. Also note that having programs in advance will only be allowed if that member s membership covers the advance quarters
4 The donation fee will be £1 per program and is not refundable Cheques and Postal Orders are to be made out to the 'U K Atari Computer Owners Club
5 Members must send in a blank cassette or diskette for the chosen programs to be recorded on
6 The 'DONATION SCHEME" is only open to club members

Finally I would like to point out that some people omit to include return postage when donating to the library so please do not forget to include 30p worth of stamps to cover this

THE LIBRARY SOFTWARE SERVICE IS FOR MEMBERS ONLY

LIBRARY SOFTWARE TITLES

Listed below are the software titles received by members for inclusion in the library since issue eleven was published As the library now contains over 300 programs it is getting a bit too large to keep on printing the entire list Eventually it would probably take over the whole magazine and there would be no room left for the articles and program listings. For those of you who are new members and do not know what is available from the library then send for a photocopy of the complete list which is available from the librarian. There is a small charge for this service to cover photocopying costs If you would like a list please send 50p and a S.A.E. for return.

Gemee

ACE OF CLUBS
by Steve Tnce – St. Neots
Pinball game designed using Pinball Construction Set
Runs in 48K min. Disk only

ANOTHER BORING SPACE INVADERS GAME
by Bob Askew – Northampton
This is the authors opinion of his work, in actual fact it's rather good!
Runs in 32K Cassette or Disk min.

SNAKE
by M Maestranzi – Finchley
Move your snake around the screen eating up the fidbits.
Runs in 16K Cassette or Disk min.

SPACE CIRCUIT
by Paul Rixon – Shefford.
Move your space ship around the circuit 4 levels of play
Runs in 32K Cassette or Disk min.

Adventure Geme

MINI ADVENTURE
by Steve Hillen – Southend
Find the hidden clues and make your escape
Runs in 16K Cassette or Disk min.

Home Entertainment

DOMINOES
by Dick Fake – Edgware.
Play a game of dominoes against the computer Cassette users should delete the Character Font i e lines 5 32000 and 32010
Runs in 32K Cassette or Disk min.

EZEE MUSIC
by Paul Rixon – Shefford.
Electronic music keyboard with a lot of features.
Runs in 32K Cassette or Disk min.

QUIZMASTER
by K. Vaughan – Old Basing.
Quiz game can add your own questions.
Runs in 32K Cassette or Disk min.

Demos

FUJIBOINK
author unknown.
Bouncing Fuji symbol demo (got to be seen to be believed) Also Fujispin and Scroll Demo included.
Runs in 48K min Disk only
Requires one side of a disk.

Utilities

EPSON GRAPHICS DUMP
by J P Crackett – Choppington
Dumps GR.8+16 screen to an Epson printer (tested on RX80) MAC/65 source code included
Runs in 16K min. Disk only

INLAY 2
by Dave Leyshon – Cambridge.
Design your own cassette inlays and print them on your 1020
Runs in 32K Cassette or Disk min

LABEL 1020
by Dave Leyshon – Cambridge.
Prints all sorts of labels on a 1020 plotter
Runs in 32K Cassette or Disk min.

LIST 1020
by Dave Leyshon – Cambridge.
Print listed programs to your 1020
Runs in 16K Cassette or Disk min

****** STAR PROGRAM ******
MAGAZINE DATABASE
by Alan J Palmer – Basingstoke.
Excellent program allows you to create a database of magazine articles includes print option (tested on RX80F/T+)
Runs in 48K min Disk only
Requires one side of a disk.
****

MULTI LABEL MAKER
by C P Weldon – Tenterden.
With this program you can print out disk labels tape labels, address labels (89 x 24mm) Epson FX or compatible
Runs in 32K Cassette or Disk min.

RANDOM CODE GENERATOR
by G Berry – Wakefield.
This program generates code numbers randomly
Runs in 16K Cassette or Disk min.

VATCALC
by Noel Davies – Telford.
Simple program for working out the VAT content of a price
Runs in 16K Cassette or Disk min.

TOP TEN

1	(–)	Homa FM	Mika Barnerd
2	(–)	Cad/Cam	Jack Oilchriat & Phil Havene
3	(–)	Mastera DOS	Metthaw Tydaman
4	(3)	Usercomp	Trevor Skeggs
5	(4)	Chase	Grahame Fairall
6	(–)	Multi	Mika Barnard
7	(–)	Skyplot	R. Barkley
8	(1)	Composed Writer	Larry Farmer
8	(–)	DOS B	Mika Barnard
10	(–)	Super Index	S. J. Davies

CP/M Emulator for ST

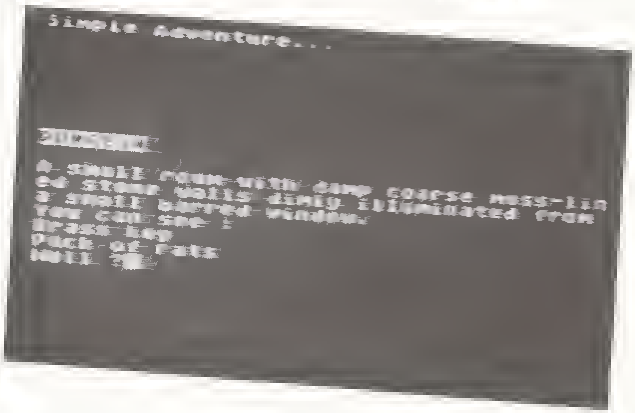
The Atari Corporation has released a CP/M 2 2 Emulator for their ST range. It comprises of 2 disks with some of its documentation in German, some in 'Wordstar' format, some as a '1st Word' letter and some as ASCII text files. Hardly in a fit state for the 'end user'

Mikes Computer Store, Westcliff in Essex (Tel no 0702 332554) have available a revamped package. All the German text has been placed in a folder on its own on the TOS disk, all English text has been corrected and placed in an ASCII document for ease of viewing in a wordprocessor of your choice In addition, a few public domain CP/M utilities have been added (before there wasn't even a decent file copying utility). And on top of this a 14 page manual has been printed to overcome the usual Atari problem – the manual was a document on the CP/M utilities disk to read the manual it was necessary to run the Emulator to run the Emulator you need to read the manual (a definite chicken and egg situation)

The improved package containing a CP/M Z80 Emulator disk, a CP/M Utility disk and the 14 page manual costs just £6 50, which includes P & P and VAT

Adventure into the ATARI

by Steve Hillen



Well, I've just got up to date with Atari matters and I see that Analog is running an adventure writing tutorial very similar to that which I had in mind. Nevertheless, here's a few suggestions for writing your own game.

In designing an adventure, the first thing to do is to decide what scenario to have and roughly outline the problems that will have to be solved, and the best way of doing this is to draw a map. My map of a very simple dungeon-type adventure is shown in Figure 1, with room names, room numbers, directions and objects being shown.

From this map, the data for the game can be drawn up. The method I used for coding was to start room numbering from 1, and have 6 pieces of data for each room. The data is the number of the room that you arrive in if you travel in one of the 6 main directions (N, S, W, E, U, D), 0 being used if there is no exit in that direction. The data is preceded by 6 dummy zeroes for the 6 directions from the imaginary room 0. For example, in line 2302, moving South (2nd bit of data) takes you to room 3.

```
PK 2299 REM Room data
PK 2300 DATA 0,0,0,0,0,0
PD 2301 DATA 0,0,0,0,0,0
AY 2302 DATA 0,3,0,0,1,0
XP 2303 DATA 2,0,4,7,0,0
UN 2304 DATA 0,0,5,3,0,0
DE 2305 DATA 0,0,0,0,0,0
SA 2306 DATA 4,0,0,0,0,0
GF 2307 DATA 0,0,3,0,0,0
```

Next, the room names must be listed, one per line in data statements, on consecutive lines, as follows:

```
WV 2000 REM Room names
BV 2001 DATA DUNGEON
JU 2002 DATA SEWER
DU 2003 DATA SEWER FORK
ND 2004 DATA KITCHEN
ST 2005 DATA CORRIDOR
FY 2006 DATA CUPBOARD
AK 2007 DATA OUTER WALL
```

The same procedure is used for the room descriptions, although you must remember that you cannot use commas in this text.

```
EX 2100 REM Room descriptions
PV 2101 DATA A small dank dark cold cell,
GP 2102 DATA An awfully smelly passage wi
   th moss- lined walls.
DL 2103 DATA The sewer bifurcates here.
GW 2104 DATA This kitchen is covered in r
   ed blood- stains. Faint noises come fr
   om the    west.
RD 2105 DATA Rotting tapestries line the
   walls.
ZP 2106 DATA This pokey cupboard is almos
   t empty.
QP 2107 DATA You are high up on the outer
    battle- ments overlooking a fatal dr
   op.
```

Finally, the objects that you can interact with must be listed. I only used 4 in this game, but keep all those objects that are fixed to the end of the list, and all those that you can pick up to the beginning.

```
HA 2200 REM Nouns
VF 2201 DATA A brass key
RB 2202 DATA A wooden ladder
KA 2203 DATA A pack of rats
PX 2204 DATA An oak door
```

This forms the major part of the data for the adventure game. The next section of data is two machine code routines which are included to speed up execution and also save memory.

```
UN 2400 REM Machine code data
XQ 2401 DATA 104,104,133,204,104,133,203,
   104,104,170,104,104,160,0,132,213,200,
   209,203,240,6,200,202,208,248
KO 2402 DATA 160,0,132,212,96,216,104,104
   ,133,204,104,133,203,104,133,206,104,1
   33,205,104,104,133,207,162,0
GG 2403 DATA 134,213,232,160,0,177,203,20
   9,205,208,9,200,192,3,144,245,134,212,
   96,165,205,24,105,3,133
MS 2404 DATA 205,144,2,230,206,232,228,20
   7,144,224,162,0,240,233
```

Now that we have most of the data, it is time to set it up in memory. Line 2505 sets up the following variables specific to this game:

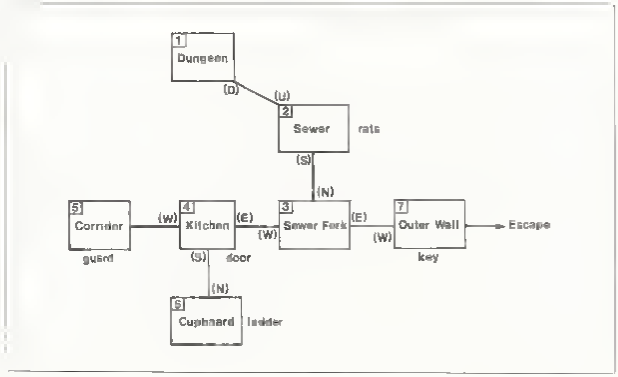


Figure 1.

NVERB – number of recognised verbs
NNOUN – number of recognised nouns
NROOM – total number of rooms.
NCMND – number of single letter commands to be understood

Line 2510 sets up the strings Note that character strings are used to save memory

V$ – contains abbreviated verbs.
N$ – contains abbreviated nouns
OBJ$ – contains locations where objects are to be found.
S$ – contains short commands – N, S, W, E, U, D and I for inventory
A$ – general purpose
K$ – input string for getting your response.
CL$ – the 6 directions for your current location.
M$ – containing all the directions for all the rooms
ST$ – a status string, used for the locked/unlocked door

The next 2 lines set up M$ with the room data, including the dummy 6 zeroes for room 0, and also poke the machine code into page 6

M1 is a character search routine, called by

X=USR(M1, adr-1, n, ch)

Where adr is the address to start the search, n is the number of bytes to search (less than 255), and ch is the code of character to be searched for

M2 is the noun/verb routine which searches for a match, only the first 3 letters of the noun or verb being significant, i e LOOKING will be interpreted as LOO or 'look'

```
QI 2500 REM Sat up data atc.
TJ 2505 NVEAB=7:ANOUN=4:RAOOM=7:NCRND=7
QD 2510 DIM V$(RVERB*3),N$(RROUR*3),OBJ$(
   NNOUN),B$(NCMAO),A$(120),K$(20),CL$(6)
   ,R$((RROOR+1)*6),ST$(1)
AL 2515 AESTOAE 2300:FOA A=1 TO (NAOOM+1)
   *6:READ O:R$(A,A)=CHR$(O):NEXT A
VP 2520 AESTORE 2400:FOR A=0 TO 69:AEAD O
   :POKE 1536+A,O:NEXT A:M1=1536:M2=M1+30
```

The next section sets up the verbs and nouns, only the first three letters of each word being significant, (e V$ stands for the following 7 verbs
take, get, drop, open, unlock, look, jump

Obviously, this method cannot distinguish between say CABIN and CABLES, which is a limitation

Also the single letter commands are put into S$, and the status of the door is set to locked (L)

Finally, the 4 objects are placed in the appropriate starting rooms, i.e. object 1, the key is placed in room 7, etc.

```
XA 2530 V$="TAKGETDAOOPEUNLLOOJUN"
ZN 2540 A$="KEYLADRATOOO"
CA 2550 G$="NSNEUOI":ST$="L"
XK 2560 FOR A=1 TO ANOUA:AEAD O:OBJ$(A,A)
   =CHA$(O):NEXT A:DATA 7,6,2,4
OA 2565 FOA A=1 TO 10:K$(A,A)=" ":NEXT A
```

Thus section sets the starting room to 1, the time counter T to zero, and fills up CL$ Commonly referenced line numbers are given a name

NEWROOM – New room procedure.
NCHECK – Checks that the input noun is valid
EXIT – Every valid move ends up here.
SHOB – Show all visible objects.

Finally, the details of the first room are printed, and all the setting up has been completed

```
RF 2570 CL=1:T=0:AOB=ADR(G$)-1:ADO=ADA(OB
   J$)-1:CL$=M$(CL*6,CL*6+5)
UQ 2580 LET NERAOOR=200:NCHECK=270:EXIT=3
   00:SHOB=230
TF 2600 GRAPHICS 0:POKE 710,144:? "SimpIa
    Adventura...";? :? :? :? :? :? :GOSUB
    NERROOR+10
AP 2610 RETUAN
```

Obviously, this setting up section must be called early on in the program.

```
IL 10 GOSUB 2500:GOTO 30
```

The following 3 lines deal with any input that is not understood. It chooses one of two error messages to avoid being too boring

```
MS 20 GOTO 21+IRT(RRO(0)*2)
KO 21 ? "Eh ?":GOTO 30
RT 22 ? "Pardon ?"
```

Line 30 is the start of the parser routine which interprets your input and sends the computer to the relevant section of the program

Line 30 prints the prompt and gets the input into K$ If it was not a single letter command then control is passed to line 140, otherwise the character search routine is called to look for a match between the typed letter and the list of valid single letter commands

```
FJ 30 ? :? "Rall ";:TRAP 20:INPUT K$:IF K
   $="" TAEA 20
CZ 40 IF LEN(K$)<>1 THEA 140
DL 50 C=USR(A1,AOB,ACMND,ASC(K$(1,1))):IF
    ROT C THER 20
```

Line 60 passes control to line 120 if a direction was typed If you asked for an inventory, S is set to zero, i.e search for objects in location zero, in your possession, and the subroutine SHOB is called.

```
MO 60 IF C<7 THEN 120
CE 70 ? "You are carrying :":? :X=0:S=0
AZ 80 GOSUB SHO0:GOTO EXIT
```

Line 120 looks in CL$ for an exit to your current room If the entry is zero, then there is no exit in that direction If there is a number there, then the NEWROOM procedure is called to change the current room

```
CO 120 B=ASC(CL$(C,C)):IF  ROT B THEN ? "
   You can't go that way.":GOTO EXIT
NF 130 GOSUB AEAADON:GOTO EXIT
```

Line 140 is the first part of the section that deals with VERB-NOUN input. It searches for the space between the two words. N and V are then calculated, being zero if the word typed in does not match one of those in N$ or V$. Note that if the verb is unrecognised, that is an immediate error, but if the noun is nonsense, it does not always matter, e g the command JUMP does not use a noun. By now, the input has been reduced to 2 numbers, referring to the position of the matched verb and noun within the two condensed strings V$ and N$ The last part of this section just transfers control to the relevant verb handling routine Each verb is dealt with separately

```
CC 140 S=USR(R1,AOR(K$)-1,15,ASC(" ")):IF
    NOT S THER 20
JB 150 N=USR(R2,ADR(K$)+S,ADA(R$),NNOUN+1
   ):V=USR(N2,ADA(K$),ADA(V$),NVEAB+1):IF
    NOT V THER 20
AT 160 GOTO 200+200*V
```

Line 200 is the start of various subroutines, which are placed near the top of the program in order to increase speed Line 200 is the new room routine, and moves the old room back into M$ and then transfers the new room from M$ into CL$ In this way, M$ reflects the current state of all the rooms Line 210 is the line that accesses all the room names and descriptions The data pointer is set to the correct line by using the RESTORE statement, the text is read into A$, and then printed. It is so that the object and room numbers tie into the data that the descriptions, etc. must be on consecutive line numbers. Lines 220-240 form a routine that does a repeated search down the length of OBJ$ in order to find all the objects present in that location If any are found, their line numbers are calculated and the text is retrieved.

```
LA 200 M$(CL*6+1,CL*6+6)=CL$:CL=S:CL$=M$(
   CL*6+1,CL*6+6)
VY 210 RESTORE 2000+CL:READ A$:? :? A$:AE
   STORE 2100+CL:READ A$:? :? A$:? "You c
   an see ":X=0:S=CL:GOSUB BAOO:AETUAN
IA 230 C=USA(A1,ADO+X,4-X,S)
JT 240 IF C THEN RESTORE 2200+X+C:READ A$
   :? A$:X=X+C:IF X<AAOUR THER 230
XK 250 IF  NOT X THEN ? "Nothing."
ZK 260 RETURR
```

This little bit just checks for a valid noun, returning if N is non zero

```
TO 270 IF  NOT N THEN POP :GOTO 20
20 280 RETURN
```

The last subroutine deals with time and a few odds and ends If time = 3 then an exit appears in the cell, and if at any time you stray into room 5, the guard captures you.

```
OF 300 T=T+1:IF T=3 THEN ? "Your eyes ada
   pt to the dark. You      notice a smal
   l hole leading downwards."
YV 310 IF T=3 THEN CL$(4,6)=CHR$(2)
UN 320 IF CL=5 THEN ? "A guard catches an
   d kills you.":END
BE 330 GOTO 30
```

This last section is the place where most of the action takes place, the verb section Each verb has its own self-contained area in which all the responses to any command containing that verb must be programmed. This may sound like a tall order, but it is actually surprisingly simple

The first two verbs, get and take, have the same meaning, so if the program arrives at line 400, it just falls through and treats the verb as though it were 'take'

The first thing to be done is to check that the noun is valid. The next question is – do you already have that object? A quick look at OBJ$ reveals the answer The next question is – is that object in the room? Again a look at OBJ$ (line 630) gives the answer Finally before you actually take the object, is the object possible to take? If N>3, i e if the object in question is the door, then you can't. Otherwise the object is placed in your possession by changing the relevant data in OBJ$ to zero

```
LN 400 REM GET/TAKE
LP 600 REM GET/TAKE
PT 610 GOSUB NCHECK
EN 620 IF  NOT RSC(OBJ$(N,N1) THEN ? "You
   already have that !":GOTO EXIT
BE 630 IF ASC(OBJ$(N,N1)<>CL THEN ? "It's
   not here !":GOTO EXIT
HN 640 IF N>3 THEN ? "You can't take that
   !":GOTO EXIT
ZF 650 ? "Ok.":OBJ$(N,N)=CHR$(0):GOTO EXI
   T
```

The next verb, drop, follows a very similar procedure In this case, the question asked is – do you actually own that object? Also, a special case is if you drop the ladder in room 7 then you escape and finish the game. Otherwise, the dropped object has the entry in OBJ$ changed from zero (inventory) to the room number

```
NR 600 REM DROP
PV 610 GOSUB NCHECK
HT 620 IF RSC(OBJ$(N,N1) THEN ? "You don'
   t have that !":GOTO EXIT
RP 630 IF N=2 AND CL=7 THEN 850
```

```
DP 640 ? "Ok.":OBJ$(N,N)=CHR$(CL):GOTO EX
   IT
MR 650 ? "The ladder reaches the ground.
   You   climb down and escape !":? "You
    took "|T|" moves.":END
```

Open is the next verb Firstly, the object must be in your possession or in the room Trying to open anything but the door (object 4) is nonsense Also, the door must be unlocked for you to open it, which is checked by looking at ST$ Finally, if the door is unlocked, then an exit appears in the room leading to room 6

```
YR 1000 REM OPEN
ER 1010 GOSUB NCHECK
YW 1020 IF RSC(OBJ$(N,N))<>CL OR  NOT RSC
   (OBJ$(N,N)) THEN ? "It's not here !":G
   OTO EXIT
NK 1030 IF N<>4 THEN ? "Eh ?":GOTO EXIT
YG 1040 IF ST$(1,1)="L" THEN ? "It's lock
   ed !":GOTO EXIT
KV 1050 CL$(2,2)=CHR$(6):? "The door opens
   ...":GOTO EXIT
```

Unlock is the fourth verb, and firstly, the object you are trying to unlock is searched for If it is not in the room nor in your possession then you cannot unlock anything Next the object is checked to be the door It doesn't make sense to unlock anything else in the game Next the door is checked to be locked – if it's already unlocked you cannot unlock it again Finally, the key is checked to be in your possession and if it is, then the door is unlocked and the status of ST$ is changed.

```
EG 1200 REM UNLOCK
EV 1210 GOSUB NCHECK
ZA 1220 IF RSC(OBJ$(N,N))<>CL OR  NOT ASC
   (OBJ$(N,N)) THEN ? "It's not here !":G
   OTO EXIT
NO 1230 IF N<>4 THEN ? "Eh ?":GOTO EXIT
NN 1240 IF ST$(1,1)="U" THEN ? "It's alre
   ady unlocked !":GOTO EXIT
FN 1250 IF ASC(OBJ$(1,1))<>0 THEN ? "You
   don't have a key !":GOTO EXIT
SC 1260 ST$(1,1)="U":? "The door is unloc
   ked.":GOTO EXIT
```

Look is a verb that does not require a noun in this case The room description is printed again.

```
ZN 1400 REM LOOK
RK 1410 GOSUB NEWROOM+10:GOTO EXIT
```

Finally, the last verb is jump This does not need a noun either, so NCHECK is not called Unless you are at the outer wall, this command has little effect.

```
OJ 1600 REM JUMP
BU 1610 IF CL<>7 THEN ? "Whaaaaa !":GOTO
   EXIT
CA 1620 ? "You see the ground rushing tow
   ards    you. It looks very hard.":FOR
   R=1 TO 150:NEXT As? "SPLAT !":END
```

I hope that this explanation has not been too difficult to follow All the separate sections of this program can be typed in and run, and you should play around with the game to see how it performs This really is a very simplistic edventure, but I'm sure you can see how the general structure and routines can be extended to produce pretty large games.

Unless there are any better suggestions mailed to me over the next few issues I shall deal with better parsers that understand whole sentences, methods of compressing data so that larger games can be written, and flashier screen displays

Unfortunataly, the Q+A section of this column seems to have died a natural death, so apologies to all those who have written in with unanswered quenes on adventures

Bits and Bytes

One of the functions missing from the Atari Basic cartridge is the ability to do bit comparisons directly. Suppose that you have a value read in from a memory location using the PEEK command. The byte you will have read is composed of 8 bits, which in combination can give a value in the range of 0 255. It is often necessary for a program to know whether a particular bit is set to 0 or 1, regardless of the other bits. The following routine will, when called by a GOSUB 1000 statement, generate a number array which has eight elements (0 to 7) each corresponding to one of the eight bits. For those of you not familiar with arrays, you look at its elements individually as follows -

B(0) for bit 0
B(1) for bit 1
B(2) for bit 2
B(3) for bit 3
etc

To use the routine you need only include lines 10 to 30 and 1000 to 1040, lines 100 to 200 merely demonstrate how to use the routine. Lines 20 and 30 create an array whose elements contain the value of 2 raised to the element number (i.e. 1,2,4,8,16,32,64,128). These represent the values of the corresponding bits, and are calculated and stored in the other array at the start of the program to avoid calculating them each time the subroutine is used, since exponential calculations are VERY time consuming!

Before calling the subroutine your program should put the number to be examined into the variable N, but remember that this will be destroyed by the routine. The elements of array B will be 0 if the corresponding bit is 0, and 1 if it is 1.

```
10 DIM B(7),N(7)
20 FOR I=0 TO 7
30 N(I)=INT(2^I+0.1):NEXT I
100 PRINT "Number...";:INPUT N
110 PRINT N;"= ";:GOSUB 1000
120 FOR I=7 TO 0 STEP -1
130 PRINT B(I);" ";:NEXT I
200 GOTO 100
1000 REM .....Bit Nap Routine.....
1010 FOR I=7 TO 0 STEP -1
1020 B(I)=0
1030 IF N>=N(I) THEN N=N-N(I):B(I)=1
1040 NEXT I:RETURN
```


ATARI 400/600/800/800XL/XE

Why not compliment the greatest home computer on the market by obtaining membership with **MIDLAND GAMES LIBRARY.**
The first, biggest and best Atari library on the market.
MIDLAND GAMES LIBRARY: The company who over three years ago first conceived the use of a software library.
Now over 1,300 titles on disk, cassette and Rom (games, business, educational and utilities). Often purchasing popular programs in multiples of five or six to give all our members a fair chance. Always adding approximately 40 new programs monthly.
Over 1 500 very satisfied members, many as far away as Iceland, Sweden, West Germany, Holland, Denmark and Eire.
A fast, efficient and friendly service. Requests are attended to immediately virtually assuring you a 24 hour return of service. Regular newsletters and program updates plus other inter-club activities. **FREE GIFT for members joining us over New Year period. Send large s.a.e. for full details to:**

MIDLAND GAMES LIBRARY
48 Readway, Bishops Cleeve, Cheltenham, Glos
Tel. 0242-67-4960 9.30 - 4.30
All our programs are originals with full documentation
*Please do not confuse **MIDLAND GAMES LIBRARY** with Midland Computers we are in no way connected or responsible for them*

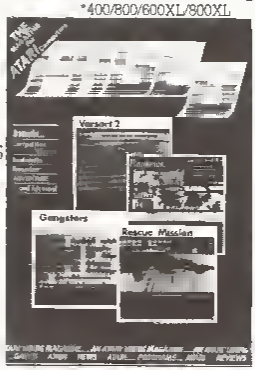
PAGE 6 THE MAGAZINE FOR ALL ATARI COMPUTER* OWNERS

*400/800/600XL/800XL

ATARI

- NEWS
- REVIEWS
- TUTORIALS
- UTILITIES
- HINTS & TIPS

plus more



THE BEST PROGRAM LISTINGS from
- U.S.A.
- U.K.
- AUSTRALIA
- PUBLIC DOMAIN SOFTWARE LIBRARY
- SPECIAL OFFERS

PAGE 6 is published bi-monthly.
Annual Subscription is £7.00. Send TODAY to:

PAGE 6, P.O. BOX 54, STAFFORD, ST16 1DR

Tel. 0785 41153

2 BIT SYSTEMS
REPLAY
The Complete Sampling System

REPLAY allows you to produce high quality digital sound through your TV or monitor.

Features:
- Runs on an Atari 400/800/600XL/800XL/130XE with 48K or more
- Sampling rate selectable from 6KHz to 21KHz.
- Allows samples to be used in your own Basic programs
- REPLAY hardware plugs into cartridge slot requires no user memory
- Cartridge required for recording only, not playback
- Records from a cassette recorder or HIFI.
- Software supplied on disc or cassette
- Also included in the REPLAY package (also available separately):

1. **DIGIDRUM** - Digital Drum sequencer (no hardware required) uses real drum samples. 8 realistic drum sounds are provided and the program allows you to make your own rhythms using all of the drums.
2. **DIGISYNTH** - Simple sample sequencer (no hardware required) allows tunes to be made from your own samples (play tunes with dogbarks, guitars, voices etc).

Prices:
REPLAY system (Replay cartridge, Sampling software, User programs, Digidrum and Digisynth) ONLY **£39.95**
Digidrum and Digisynth on disc or cassette ONLY **£4.95**
TRADE ENQUIRIES WELCOME

ORDER FORM:
Please send me _____ REPLAY packages.
Please send me _____ DIGIDRUM/DIGISYNTH packages on DISK/TAPE (delete as applicable).
I enclose a Cheque/P.O. for £ _____ (payable to 2 Bit Systems)
Name .....
Address .....
.....
.....
Postcode .....
Send to: 2 Bit Systems, 44 Manvers Road, Dunstable, Beds. LU5 4EG

Take out a year's subscription to ATARI USER and we will send you this top-selling game absolutely FREE!

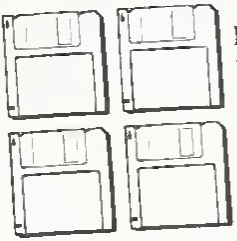
Cassette only Suitable for all Atan 8-bit machines.

Caught on the surface of a nuclear devastated planet, you have only minutes to return underground before your radiation shield decays. In any other car you would stand no chance – in The Last V8 survival is possible... just!

This greot game is the challenge of o lifetime! It feotures:

- Super-smooth 4-directional scrolling
- Top quality scene-setting graphics
- Superb theme music
- Detailed instrument panel readout
- Gripping arcade action

SPECIAL SUBSCRIPTION OFFER



For Atari ST users we can offer FOUR 3½" single sided discs for the unbeatable price of £5.99!

Only avoilable when occompanied by o subscription order.

All prices include postage, packing and VAT
Overseas orders despatched by Air mail.

Please enter number required in box (✓) £ p

Annual subscription

UK & EIRE £12 (sterling only)	7001	
Europe £15	7003	
Overseas (Airmail) £30	7004	
Tick box if renewal		

Commence with _____ issue

Exclusive offers for subscribers

The Last V8 (tape)	Free (UK only)*	7071
4.3½" discs	£5.99*	7068

Add £2 for Europe/Overseas

**These offers only available if accompanied by a subscription order*

Payment please indicate method (✓)

☐ Access/Mastercard/Eurocard Barclaycard/Visa TOTAL

☐ Cheque/PO made payable to Database Publications Ltd

Credit card expiry date /

Name _____

Address _____

_____

Signed _____

Send to *Atari User*, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY

Please allow 28 days for delivery

Order at any time of the day or night

Orders by Prestel	Telephone Orders:	Enquiries only
Key *89, then 614568383	061-429 7931	061-480 0171 9am-5pm

Don't forget to give your name, address and credit card number MON

CRACKING THE CODE

by Keith Mayhew Part Eight

We finished last time with a program which accessed a graphics 8 screen, randomly setting pixels on or off. This time we will start off with a description of a more advanced program which allows drawing on a graphics 8 screen with a joystick.

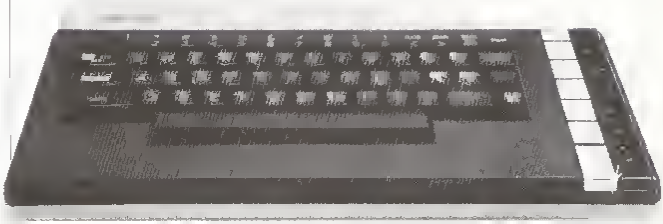
The Program

The assembly language for the drawing program is shown in Listing 1, the assembled code can be loaded by the BASIC program shown in Listing 2 Assuming the code has now been loaded, it can be run by typing the following line into BASIC

GRAPHICS 24 X=USR(20480)

Before we start looking at the program in detail, here is a short description of what the program actually does A cursor is displayed on the screen in the form of a cross-hair which can be moved around with a joystick. If the joystick button is pressed then points will be plotted on the screen under the cross There are four keys on the keyboard which are looked at, they are the 'E' key which causes any further drawing to erase when the button is pressed, the 'D' key will switch back to normal drawing mode The keys 'S' and 'F' will change the speed of drawing from slow to fast respectively, initially slow is selected Lastly, if one of the START, SELECT or OPTION keys are pressed then the program will return back to BASIC

Now we know what the program does we can start looking at the assembly language listing The first five equates define system locations The new locations here are 'STICK0' which holds the up/down/left/right information for joystick 0, STRIG0 which holds the state of the button for joystick 0, and CH which holds a code for the most recent key pressed Next is a list of the program's variables starting at CB hex. PLTMOD holds a value which determines how a point is plotted, if it is 0 then the point will be erased, if it is 1 then the point will be drawn and if it is anything else it will cause the point to be inverted These plotting mode values are defined from line 300 to 320 as ERA, DRW and INV respectively COLOUR is used to hold the current plotting mode used for drawing with the joystick and is either 'erase' or 'draw' XPOS and



YPOS hold the current pixel position, note that the X position is a two byte variable because there are 320 pixels across a graphics 8 screen The maximum values for X and Y are defined as XMAX and YMAX on lines 260 and 270 CURSON is set to FF hex when the cursor is displayed and to zero when it is turned 'off' DELVAL holds the delay value which determines how fast the cursor can move around the screen, the two speeds used are defined as SLOW and FAST on lines 280 and 290 The last few variables are used within subroutines only and their use will be seen later

The program starts by initialising the X and Y positions to half their maximum values. The colour is set to 'draw' and the speed to 'slow' A call to cursor draws the cross-hair in the initial position and then the CURSON flag is set to FF hex to indicate the cursor is on. This value is also stored in CH which clears any key code which might have been stored there

The main loop starts by looking at the state of the consol keys, if any are down then the program exits If not, then the key code is loaded from CH and compared against the values of 3A, 2A, 3E and 38 hex which are the codes for the upper case letters D, E, S and F respectively, they are not the ASCII values for those letters but an awkward internal code for the keyboard If any of these keys are detected then the appropriate value is stored in COLOUR or DELVAL If one of the keys had been pressed then FF hex is stored back afterwards so that the key is not detected the next time around the loop

The value of STICK0 is examined next, if it is equal to 0F hex, then the joystick has not been moved from its centre position If it has been moved then the cursor is turned off by calling the cursor routine again and the move routine is called to move the cursor in the appropriate direction The button state of the joystick is then examined If it is up it has the value of one and zero if it is pressed If the button is down then the cursor is turned off if it is currently on, this would happen if the cursor had not been moved The plotting mode is then set to the current colour value and the point is plotted. Lastly, if the cursor is on then neither the joystick had been moved or the button pressed and a jump is made back to the start of the loop Otherwise the cursor is turned back on, a delay is generated and it then jumps back to the start of the loop

If one of the consol keys had been pressed then the program jumps to 'EXIT' which resets any key code which might have been generated and returns back to BASIC We will now finish with a study of the four subroutines called by the main program, they are MOVE, CURSOR, PLOT and DELAY

```
0100 SRVMSC  =    $58     ;Screen palatar.
0110 STICK0  =    $0278   ;Joystick 0 value.
0120 STRIG0  =    $0284   ;Joystick 0 button value.
0130 CH      =    $02FC   ;Key press value.
0140 CONSOL  =    $D01F   ;Consol keys.
0150         *=   $CB     ;Set to page zero area.
0160 PZERO   *=   *+2     ;Pointer to access screen.
0170 TMPLOT  *=   *+2     ;Temporary for screen access.
0180 PLTMOD  *=   *+1     ;Plotting mode.
0190 COLOUR  *=   *+1     ;Colour value.
0200 XPOS    *=   *+2     ;Current X position.
0210 YPOS    *=   *+1     ;Current Y position.
0220 CURSON  *=   *+1     ;Cursor on/off flag.
0230 TMPCRS  *=   *+1     ;Temporary to hold X or Y.
0240 DELVAL  *=   *+1     ;Delay value.
0250 COUNT   *=   *+1     ;Count value.
0260 XMAX    =    319     ;Maximum X pixel number.
0270 YMAX    =    191     ;Maximum Y pixel number.
0280 SLOW    =    $10     ;Slow speed delay value.
0290 FAST    =    1       ;Fast speed delay value.
0300 ERR     =    0       ;Erase colour.
0310 DRW     =    1       ;Drawing colour.
0320 INV     =    $FF     ;Inverse mode.
0330         *=   $5000   ;Start program at 20K.
0340         PLA          ;Clean up stack.
0350         LDA  SXMAX/2 ;Load half of maximum X value.
0360         STA  XPOS    ;Save as start X position.
0370         LDY  #0      ;Set high byte of
0380         STA  XPOS+1  ; X position to zero.
0390         LDA  YMAX/2  ;Load half of maximum Y value.
0400         STA  YPOS    ;Save as start Y position.
0410         LDY  $DRW    ;Set initial colour
0420         STY  COLOUR  ; to 'draw'.
0430         LDA  #SLOW   ;Set initial delay
0440         STA  DELVAL  ; to 'slow'.
0450         JSR  CURSOR  ;Draw the cursor.
0460         LDA  #$FF    ;Set cursor flag
0470         STA  CURSON  ; to 'on'.
0480         STA  CH      ;Reset key code value.
0490 LOOP    LDA  CONSOL  ;Set consol keys.
0500         CMP  #7      ;If any are down
0510         BNE  EXIT    ; then exit.
0520         LDA  CH      ;Set key code.
0530         CMP  #$FF    ;If no key
0540         BEQ  TSTICK  ; then skip key tests.
0550         CMP  #"R     ;See if it is code for 'R'.
0560         BNE  TSTERA  ;No, then try next.
0570         LDY  #DRW    ;Set colour
0580         STA  COLOUR  ; to 'draw'.
0590         JMP  RSETCH  ;Skip rest of key tests.
0600 TSTERA  CMP  #"E     ;See if 'E'.
0610         BNE  TSTSLR  ;No, then try next.
0620         LDA  #ERA    ;Set colour
0630         STA  COLOUR  ; to 'erase'.
0640         JMP  RSETCH  ;Skip rest.
0650 TSTSLN  CMP  #"S     ;See if 'S'.
0660         BNE  TSTFST  ;No, then try next.
0670         LDA  #SLOW   ;Set speed
0680         STA  DELVAL  ; to 'slow'.
0690         JMP  RSETCH  ;Skip last test.
0700 TSTFST  CMP  #"F     ;See if 'F'.
0710         BNE  RSETCH  ;Skip if not.
0720         LDA  #FAST   ;Set speed
0730         STA  DELVAL  ; to fast.
0740 RSETCH  LDA  #$FF    ;Reset key code.
0750         STA  CH
0760 TSTICK  LDA  STICK0  ;Get stick value.
0770         CMP  #$0F    ;If no movement
0780         BEQ  TSTBUT  ; then skip.
0790         JSR  CURSOR  ;Else turn off cursor.
0800         JSR  MOVE    ;Update X and Y position.
0810 TSTBUT  LDA  STRIG0  ;Get button value.
0820         BNE  DRWCRS  ;Skip if up.
0830         LDY  CURSOR  ;Else see if cursor is off.
0840         BEQ  PLTPRT  ; Yes, then skip.
0850         JSR  CURSOR  ;Else turn it off.
0860 PLTPRT  LDA  COLOUR  ;Get current colour.
0870         STA  PLTMOD  ;Save as plotting mode.
0880         JSR  PLOT    ;Plot the current point.
0890 DRWCRS  LDA  CURSON  ;See if cursor is on.
0900         BNE  LOOP    ;Yes then go to start.
0910         JSR  CURSOR  ;Else turn cursor back on.
0920         JSR  DELAY   ;Produce a delay.
0930         JMP  LOOP    ;Go to start.
0940 EXIT    LDA  #$FF    ;Clear any
0950         STA  CH      ; key code.
0960         RTS          ;Return.
0970 ;
0980 ;Move X and Y depending on stick value.
0990 MOVE    LDA  STICK0  ;Get stick value.
1000         ORA  #1      ;Look at 'up' bit.
1010         BNE  DOWN    ;Skip if not down.
1020         LDA  YPOS    ;Get Y value.
1030         BEQ  LEFT    ;Skip if at zero.
1040         DEC  YPOS    ;Else decrement Y.
1050         JMP  LEFT    ;Go to 'left' test.
1060 DOWN    LDA  STICK0  ;Get stick
1070         AND  #2      ; 'down' bit.
1080         BNE  LEFT    ;If not down then skip.
1090         LDA  YPOS    ;Else get Y value.
1100         CMP  #YMAX   ;If at maximum
1110         BEQ  LEFT    ; then skip.
1120         INC  YPOS    ;Else increment Y value.
1130 LEFT    LDA  STICK0  ;Get stick
1140         AND  #4      ; 'left' bit.
1150         BNE  RIGHT   ;Skip if not down.
1160         LDY  XPOS    ;Get X value.
1170         BNE  LEFT2   ;If not zero then OK.
1180         LDA  XPOS+1  ;Else if high byte is
1190         BEQ  MOVEXT  ; zero then skip rest.
1200         DEC  XPOS+1  ;Else decrement high byte as well.
1210 LEFT2   DEC  XPOS    ;Decrement low byte of X.
1220         JMP  MOVEXT  ;Skip 'right' test.
1230 RIGHT   LDA  STICK0  ;Get stick
1240         AND  #8      ; 'right' value.
1250         BNE  MOVEXT  ;Skip if not down.
1260         LDA  XPOS+1  ;Get X high byte.
1270         BEQ  RIGHT1  ;If zero OK.
1280         LDA  XPOS    ;Else if X low byte
1290         CMP  #XMAX%&#FF ; is at maximum then
1300         BEQ  MOVEXT  ; skip rest.
1310 RIGHT1  INC  XPOS    ;Increment low byte.
1320         BNE  MOVEXT  ;Skip if not zero.
1330         INC  XPOS+1  ;Else adjust high byte.
1340 MOVEXT  RTS          ;Return
1350 ;
1360 ;Draw cursor in 'inverse' mode.
1370 CURSOR  LDY  CURSON  ;Get cursor state.
1380         EOR  #$FF    ;Flip state.
1390         STA  CURSOR  ;Save back.
1400         LDA  #INV    ;Set plot mode
1410         STA  PLTMOD  ; to 'inverse'.
```

```
1420          JSR     PLOT    ;Plot current point.
1430          LDA     YPOS    ;Save Y value.
1440          STA     TMPCRS
1450          LDX     #4      ;Get count to 4 pixels.
1460          STA     COUNT
1470 CRSUP1   LDA     YPOS    ;Get Y value.
1480          BEQ     CRSOR   ;If zero then skip.
1490          DEC     YPOS    ;Else decrement it.
1500          JSR     PLOT    ;Plot point.
1510          DEC     COUNT   ;Decrement count
1520          BNE     CRSUP1  ;Continue if more.
1530 CRSOR    LDA     TMPCRS  ;Restore Y value.
1540          STA     YPOS
1550          LDA     #4      ;Reset count.
1560          STA     COUNT
1570 CRSDR1   LDA     YPOS    ;Test Y value
1580          CMP     #YMAX   ; against maximum.
1590          BEQ     CRSLT   ;Skip if equal.
1600          INC     YPOS    ;Else increment it.
1610          JSR     PLOT    ;Plot point.
1620          DEC     COUNT   ;Decrement count.
1630          BNE     CRSDR1  ;Continue if more.
1640 CRSLT    LDA     TMPCRS  ;Restore Y value.
1650          STA     YPOS
1660          LDA     XPOS    ;Save X value.
1670          STA     TMPCRS
1680          LDA     XPOS+1  ;Save X high byte
1690          PHA             ; on the stack,
1700          LDA     #4      ;Reset count.
1710          STA     COUNT
1720 CRSLT1   LDA     XPOS    ;See if X is zero.
1730          BNE     CRSLT2  ;OK if not.
1740          LDA     XPOS+1  ;Else if high byte
1750          BEQ     CRSRT   ; zero then skip.
1760          DEC     XPOS+1  ;Else decrement high byte.
1770 CRSLT2   DEC     XPOS    ;Decrement low byte.
1780          JSR     PLOT    ;Plot point.
1790          DEC     COUNT   ;Decrement count.
1800          BNE     CRSLT1  ;Continue if more.
1810 CRSRT    LDA     TMPCRS  ;Restore X value.
1820          STA     XPOS
1830          PLA             ;Get high byte and save
1840          PHA             ; another copy on the stack.
1850          STA     XPOS+1  ;Restore high byte.
1860          LDA     #4      ;Reset count.
1870          STA     COUNT
1880 CRSRT1   LDA     XPOS+1  ;Get high byte.
1890          BEQ     CRSRT2  ;OK if zero.
1900          LDA     XPOS    ;Else test low byte against
1910          CMP     #XMAX&$FF ; maximum value.
1920          BEQ     CRSEXT  ;If equal then skip.
1930 CRSRT2   INC     XPOS    ;Else increment low byte.
1940          BNE     CRSRT3  ;Skip if not zero.
1950          INC     XPOS+1  ;Increment high byte.
1960 CRSRT3   JSR     PLOT    ;Plot point.
1970          DEC     COUNT   ;Decrement count.
1980          BNE     CRSRT1  ;Continue if more.
1990 CRSEXT   LDA     TMPCRS  ;Restore X low byte.
2000          STA     XPOS
2010          PLA             ;Restore X high byte.
2020          STA     XPOS+1
2030          RTS             ;Return.
2040 ;
2050 ;Plot a point at current X and Y.
2060 PLOT     LDA     SAVRSC  ;Copy screen pointer
2070          STA     PZERO   ; into our pointer.
2080          LDA     SAVRSC+1
2090          STA     PZERO+1
2100          LDA     YPOS    ;Get Y value.
2110          STA     TMPLOT  ;Save in temporary location.
2120          LDA     #0      ;Set high byte to zero.
2130          STA     TMPLOT+1
2140          LDX     #3      ;Set for eight times...
2150 MULT8    ASL     TMPLST  ;Shift left on both bytes.
2160          ROL     TMPLOT+1
2170          DEX             ;Decrement count
2180          BNE     MULT8   ;Loop.
2190          LDA     PZERO   ;Get pointer value.
2200          CLC
2210          ADC     TMPLOT  ;Add low byte.
2220          STA     PZERO   ;Save back.
2230          LDA     PZERO+1 ;Same for high byte.
2240          ADC     TMPLOT+1
2250          STA     PZERO+1
2260          ASL     TMPLOT  ;Shift twice more
2270          ROL     TMPLOT+1 ; to get 32 times.
2280          ASL     TMPLOT
2290          ROL     TMPLOT+1
2300          LDA     PZERO   ;Add to pointer value.
2310          CLC
2320          ADC     TMPLOT
2330          STA     PZERO
2340          LDA     PZERO+1 ;And high byte...
2350          ADC     TMPLOT+1
2360          STA     PZERO+1
2370          LDA     XPOS+1  ;Get high byte of X.
2380          LSR     A       ;Shift low bit into carry.
2390          LDA     XPOS    ;Get low byte.
2400          ROR     A       ;Rotate carry in.
2410          LSR     A       ;Shift twice more to
2420          LSR     A       ; divide X by a total of 8.
2430          TAY             ;Save in Y register as index.
2440          LDA     XPOS    ;Get low byte of X.
2450          AND     #7      ;Mask to get low three bits.
2460          TAX             ;Move to X register.
2470          LDA     #$80    ;Load with high bit set.
2480 SHIFT    DEX             ;Decrement count.
2490          BMI     PLOT1   ;Skip if shifting finished.
2500          LSR     A       ;Else shift.
2510          JMP     SHIFT   ;Continue if more.
2520 PLOT1    LDX     PLTMOD  ;Get plotting mode.
2530          BEQ     ERASE   ;If zero then 'erase' mode.
2540          CPX     #DRW    ;If 'draw' mode then
2550          BEQ     DRAW    ; skip to it.
2560          EOR     (PZERO),Y ;'Exclusive-or' for 'inverse'.
2570          JMP     PLTEXT  ;Skip rest.
2580 DRAW     ORA     (PZERO),Y ; 'Or' for 'draw' mode.
2590          JMP     PLTEXT  ;Skip rest.
2600 ERASE    EOR     #$FF    ;Invert bits.
2610          AND     (PZERO),Y ; and 'and' for 'erase'.
2620 PLTEXT   STA     (PZERO),Y ;Save value back to screen.
2630          RTS             ;Return.
2640 ;
2650 ;Produce a small delay.
2660 DELAY    LDX     DELVAL  ;Get delay value.
2670 DELAY1   LDY     #$FF    ;Inner delay loop.
2680 DELAY2   DEY
2690          BNE     DELAY2  ;Loop.
2700          DEX             ;Decrement outer loop value.
2710          BNE     DELAY1  ;Loop.
2720          RTS             ;Return.
```

Listing 2

```
OZ 10 DIA HEX$(16)
TU 20 LINE=10000:TRAP 100:J=0:BTAAT=20480
VA 30 READ HEX$,CHKSUA:SUA=0
AA 40 FOA I=1 TO 15 STEP 2
ZG 50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(I+
   1,I+1))-48
KT 60 NUM=((D1-7*(D1)161)*16+(D2-7*(D2)16
   1))
LM 70 SUM=SUM+NUM:POKE START+J,NUM:J=J+1:
   NEXT I
LY 80 IF SUM=CHKSUM THEN LINE=LINE+10:GOT
   O 30
IN 90 ? "Checksue error on this line:"
VO 95 LIST LINE:END
YS 100 PRINT "Data in memory."
GM 10000 DATA 68A99F85D1A90085,1076
JG 10010 DATA 02A9F8B5D3A9D185,1121
ET 10020 DATA DBA91085D620D850,1071
NE 10030 DATA A9FF85D4B0FCD2AD,1337
KO 10040 DATA 1FD0C907D0A2A0FC,1178
TN 10050 DATA 02C9FFF02EC93AD0,1211
KX 10060 DATA B7A9D185D04C5650,760
MH 10070 DATA C92AD007AFB065D0,968
EH 10080 DATA 4C5650C93ED007AF,889
BE 10090 DATA 1865D64C5650C958,862
NY 10100 DATA D004A9D185D6A9FF,1153
IX 10110 DATA 6DFCD2AD7BB2C90F,986
KO 10120 DATA F0B620D0502D0E50,831
GJ 10130 DATA ADB4D2D00EA5D4F0,1146
FC 10140 DATA 0320D05DA5D0E5CF,1047
WF 10150 DATA 2D6151A5D4D0AA2D,987
FE 10160 DATA D050D0CC514C1F50,803
GG 10170 DATA A9FFB0FCD26BAD78,1288
IN 10180 DATA 022901D009A5D3F0,877
MG 10190 DATA 14C6D34CAD50AD78,1051
KO 10200 DATA B22902D008A5D3C9,838
KB 10210 DATA 3FFB02E6D3A5D7B0,1169
IG 10220 DATA 2904D00FA5D1D0D6,854
TP 10230 DATA A5D2F01EC6D2C6D1,1460
DT 10240 DATA 4CDA50AD78B22902,718
QY 10250 DATA D010A5D2F006A5D1,1219
WZ 10260 DATA C93FF006E6D1D0D2,1159
TE 10270 DATA E6D26BA5D449FFB5,1374
BQ 10280 DATA D4A9FFB5CF286151,1186
XB 10290 DATA A5D385D5A7B485D7,1243
CA 10300 DATA A5D3F00FC6D32861,1163
NC 10310 DATA 51C6D7D0F3A5D585,1456
DK 10320 DATA D3A9D485D7A5D3C9,1309
ZO 10330 DATA 0FFB09E6D3286151,1071
NP 10340 DATA C6D7D0F1A5D585D3,1584
JG 10350 DATA A5D1650A5D246A9,1336
JY 10360 DATA B485D7A5D1D006A5,1185
YG 10370 DATA D2F00BC6D2C6D120,1308
MT 10380 DATA 6151C6D7D0EDA5D5,1414
BM 10390 DATA 85D16848D5D2A9D4,1034
YZ 10400 DATA B5D7A5D2F006A5D1,1343
EE 10410 DATA C93FF000E6D1D0D2,1166
NK 10420 DATA E6D22B6151C6D7D0,1271
AY 10430 DATA E9A5D585D1A985D2,1480
RK 10440 DATA 68A558B5CBA55905,1072
YU 10450 DATA CCA5D385CDA90085,1228
IN 10460 DATA CEA2D306CD26CECA,1026
DA 10470 DATA D0F9A5C01865CD85,1288
CE 10480 DATA CBA5CC65CE85CCB6,1222
RA 10490 DATA CD26CE06CD26CEA5,1069
NE 10500 DATA CB1B65CD85CBA5CC,1236
BN 10510 DATA 65CE85CCA5D24AA5,1258
GN 10520 DATA D16A4A4AAAA5D129,1046
DR 10530 DATA 07AAA98DCA38044A,882
CF 10540 DATA 4CAC51A6CFF0EED,1180
VA 10550 DATA B1F08551C84CC951,888
FO 10560 DATA 11CB4CC9514FFF31,955
LU 10570 DATA CB91CB6BA6B6AAFF,1442
PT 10580 DATA BBD0FDCAD0FB60,1351
```

The subroutine 'MOVE' consists of four similar sections which load the value of the joystick and adjust the X and Y values according to the direction specified. The coding scheme used for the variable STICK0 is as follows. The upper four bits are unused and are always zero, the lower four bits are normally all ones which refers to no movement (hence the value of 0F hex was used for the comparison on line 770). These lower four bits, starting from bit 0, represent the following directions on the joystick if they are set to zero: UP, DOWN, LEFT and RIGHT. Only one or two of these bits can be set to zero at any time depending on if the joystick is at a diagonal or is pointed straight in one direction.

The direction 'up' causes the Y value to be decremented unless it is already at zero, i.e. the top of the screen, and the 'down' test is skipped because both bits could not be set at once. The 'down' direction increments the Y value unless it is already at the value of YMAX, i.e. at the bottom of the screen. The left and right directions are handled in the same way except that the X direction has a high byte which needs to be tested, but note that this byte can only take on the values of 0 or 1 depending on if the current X value is below 256 or above 255 respectively.

The 'CURSOR' routine draws the cross-hair, it consists of the current pixel and four adjacent pixels in all four directions. The value in 'CURSON' is complemented so that if the cursor was on it is now off and vice versa. The plotting mode is set to 'inverse' which causes the value of any pixel plotted to be complemented. By using the inverse mode, if the cursor is drawn once it appears on the screen and if it is drawn again it will cause it to flip back to its previous state so that it cannot be seen. This also allows the cursor to be plotted over existing pixels without having to save their values before the cursor is plotted. Drawing the cursor consists of four similar sections, these draw four pixels in the four directions from the current location, before this is done the current pixel is complemented by a call to 'PLOT' on line 1420. The first section of code plots four pixels in the upward direction, before each pixel is plotted the Y value is tested to see if it is at the top of the screen, in which case no more pixels are drawn in that direction. The next section draws four pixels downwards, but first the original value of Y is restored from the temporary location 'TMPCRS'. The cursor left and right code is similar to the up and down code except that the high byte of the X position also has to be saved and restored, this is achieved by using the stack.

The 'PLOT' routine is fundamentally the same as the routine used in Listing 1 last time. The Y value is loaded into the temporary locations TMPLOT and TMPLOT+1. This is then multiplied by 8 and added onto the value in PZERO and PZERO+1, which pointed to the start of the screen. The temporary value is then multiplied by another 4 to give 32 times and added onto the pointer so that, in effect, 40 times the Y value has been added. The next task is to find the offset into the line. As there are eight pixels to each byte we need to divide the X value by eight, this is done on lines 2370 to 2420. Note that only one shift is necessary on the high byte of X because only bit 0 is used. This byte offset is then saved in the Y register for indexing. We now have to find the appropriate bit which we wish to access within that byte. This is done by taking the low byte of the X value and masking off all but the lowest three bits. This number will now represent the bit number to be accessed (if we treat the left most bit of the byte as zero). This value is moved into the X register as a count, and the accumulator is loaded with 80 hex so that the left most bit is set. The accumulator is then shifted to the right by the number of times specified in the X register. The value of PLTMOD then determines how we place this byte onto the screen. If it is equal to 'DRW' then it is simply ORed to the screen byte and saved back, thus setting the pixel to on. To erase the pixel, PLTMOD has the value 'ERA' (zero) and the byte in the accumulator is exclusive-ORed with FF hex inverting every bit so that the bit to be erased is now set to zero. This is ANDed with the screen byte setting the appropriate bit to 0 and leaving the rest unchanged. Lastly, if we are in the 'inverted' mode then the byte is exclusive-ORed with the screen byte, thus inverting that pixel. Whichever method was used, the accumulator would hold the correct byte to be saved back to the screen at line 2620.

The last routine is 'DELAY'. This uses the X and Y registers to produce a variable delay depending on the value in

the variable 'DELVAL' The inner loop is repeated 256 times with the Y register and this is repeated with the outer loop by the number of times specified in DELVAL.

If you can follow how the program works then you are probably ready to write your own programs However, if you feel significantly lost with this program then it is time to start a little revision!

As promised last time, we will now start our detailed tour of the ATARI hardware and operating system Most programs will now be written to illustrate how to use a certain part of the machine You may find a few useful techniques which you could also apply in your own code in the process

The Memory Map

As you will be aware, there are 64K bytes of memory which the 6502 processor can address The division of this memory between the various parts of the system is usually illustrated by a 'memory map' Figure 1 shows the memory map for all the 8-bit ATARI computers, namely the 400, 800 600XL, 800XL and 130XE models All of these machines are fundamentally the same, most differences will be noted in due course

The memory map shown gives the sizes of the main areas on the right and to the left are the addresses of these areas, some of which are shown in 'K's for reference The operating system occupies the top 8K of the memory from E000 to FFFF hex and the 4K area from C000 to CFFF hex Note that the latter 4K is only present in the XL and XE range The operating system (O S) is, obviously, held in ROM so that it is always present, remember that if you try to write to ROM nothing will happen, you cannot change its contents The reason why the O S occupies the top locations of memory is that if you recall, the top six bytes of the address range are special to the 6502 Locations FFFC and FFFD hex hold the address of the routine which will be executed on power up Locations FFFA and FFFB hex hold the address of the non-maskable interrupt (NMI) routine. Locations FFFE and FFFF hex hold the address of the interrupt request (IRQ) routine The three addresses will all point to somewhere inside the O S. ROM area. The 2K area from D800 to DFFF hex is another area of ROM which contains the floating point (F P) software to handle arithmetic on such numbers The F P software can be considered as part of the operating system, however, it is functionally completely separate

The two 8K regions from 8000 to BFFF hex are where the ROM cartridges reside The idea of two separate cartridges, 'A' and 'B', comes from the old 800 model and was done mainly because it was not possible to get 16K

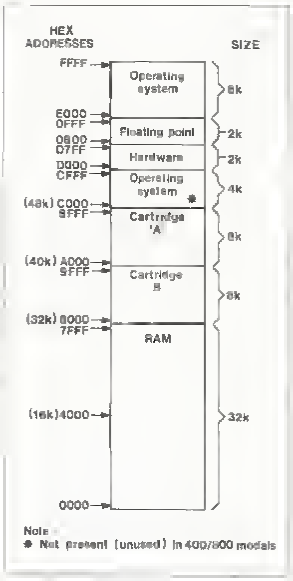


Figure 1

into one physical cartridge at the time. If an 8K cartridge is installed in a machine then it will occupy the 'A' cartridge area, if it is a 16K cartridge then it will occupy both areas 'A' and 'B' Note that ATARI BASIC is 8K in size and is installed as a cartridge automatically by the XL and XE machines as it is built inside the machine, it is deselected (effectively removed) by either inserting another cartridge to take its place or by holding down the OPTION key

The size of the RAM area is different between machines but it is always present from location zero and extends upwards Assuming you have more than 32K of memory, then if the cartridge areas are not used then the RAM will be accessible in those areas, otherwise it will retain its data but will not be accessible The same principle applies to the XL and XE machines where there is hidden RAM in the top 16K of the machine which can be made accessible.

The Hardware

We will finish this time by starting an examination of the 2K area from D000 to D7FF hex which contains the hardware We have already used two of these locations in our programs, namely CONSOL and RANDOM When accessing these 'memory' locations you are actually accessing one of the special chips of the ATARI

The areas used for the four chips are as follows. The 'GTIA' chip occupies the locations D000 to D0FF hex, the 'POKEY' chip occupies locations D200 to D2FF hex, the 'PIA' chip occupies locations D300 to D3FF hex and the 'ANTIC' chip from D400 to D4FF hex. Very few of these locations for each chip actually have a use The hardware locations on these chips are usually called registers but the way these registers can be accessed varies from register to register, the various types are as follows Some registers are read/write, that is they behave like RAM locations· others are read only, that is they act like ROM and others are write only, that is you cannot read back what is held in a register The read only and write only registers often occupy the same location, that is on reading you will access one register and on writing you will access the other Lastly there are some registers which are activated simply by writing to them, i.e the actual data written is irrelevant.

We will continue next time with our study of the hardware In the mean time, if you have not got a decent reference manual for the hardware and the operating system then it is highly recommended that you get ATARI's TECHNICAL REFERENCE NOTES which also contains a complete listing of the 400/800 operating system as well as the circuit diagrams These notes can be a little heavy going, but they are very thorough and useful A more practical guide, again recommended, is COMPUTES!'s MAPPING THE ATARI (XL/XE edition)

MONITOR ON DISK

Like the look of a program but can't find the time to key it in? You've asked the wife three times to do it for you whilst you're out at work, and she still hasn't. Or maybe you have typed it in but it won't run then why not take all the effort out of it and send for the MONITOR DISK. All the main programs in each issue of MONITOR are now available pre recorded on disk for you. They cost £4.95 which includes postage and packing, send a cheque/postal order made payable to the 'U.K. Atari Computer Owners Club' to Monitor Magazine, P.O. Box 3, Rayleigh, Essex. If you live in Europe add 50p, if outside Europe add £1.00. Please allow 28 days for delivery.

Monitor Disk 8.
Includes Quickplot, a fast Graphics 8 Plot/Drawto handler. Nightmare Reflections, an exceedingly frustrating adventure. Matchbox, improve your concentration with this memory game. Interrupts, 5 demo programs showing various uses of Interrupts.

Monitor Disk 9.
Includes Keyo, a new typing checker. Multiboot Bootbase, database program for 'Multiboot disks'. Binload, binary loads from BASIC. Happytyper, automatic line numbering. Ramdisk, for use with the 130XE. Fast Fill, a speedy shape filling utility.

Monitor Disk 10.
Includes 3D Maze, escape from the maze in time if you can. PCB Paranoia, destroy your enemies before they get you. Disk Jacket, useful program for making your own disk covers. Chase, an excellent game, not to be missed.

Monitor Disk 11
Includes Hexadecimal Code Generator, better presentation for your programs. Cracking the Code, seven mini progs from the series. RAM Talker, with a little bit of hardware and this program, you can hear your own voice, (for 400/800 only). Bonus Program HomeFM, a useful utility for use with Home Filing Manager to give quick access to data disks.

Monitor Disk 12.
Includes Another Boring Space Invaders Game, unlike its name suggests this game is rather good and lots of fun. Get Motorised, four programs for use with the circuits described in this interesting article. Mini-adventure, can you escape in one piece? Cracking the Code. BASIC listing and assembler code for a drawing program. Opening Out, five useful programs for disk drive owners.

BACK ISSUES

Previous issues of this magazine are obtainable from the club for £1 plus 30p postage each. They contain many interesting and informative articles, hints & tips, program listings for you to input, reviews and practical advice. If you have missed out send for your copies of back issues today! Please note that issues 1,2,3,4,5 & 7 are already sold out.

Issue 6.
Includes a useful tutorial showing how to print Micropainter and Versawriter pictures, also contains a terrific program demonstrating 80 characters across the screen. A new regular column for adventure enthusiasts is started to give reviews of adventure games and give hints and tips on how to play them. Part two of Cracking the Code continues with addressing modes and binary sums. The hardware design for a Light Pen is shown together with some simple programs use with it once you have built it. Fun with Art from Epyx is reviewed and some of the excellent results of using this package are shown. Programs include Planetron and a RTTY listing for use with a short wave band radio, the Atari 850 interface and a signal terminal unit (such as the Maplin TU1000).

Issue 8.
Contains a preview of the new Atari computers. Two new series start, one about how files work and the other 'Starting from Basics' for beginners. Cracking the code continues and concluding part of 'Interrupts' discusses horizontal and vertical scrolling. The adventure column includes reviews of Mask of the Sun and Sorcerer. Other reviews include Conan, Spy vs Spy, Alley Cat and Ghostbusters. Programs are Matchbox, a concentration game, Quickplot a Graphics 8 Plot/Drawto utility and Nightmare Reflections, an exceedingly frustrating adventure.

Issue 9.
Includes a RAMDISK for the 130XE as well as a review of this excellent machine. Introduction to MIDI, just what is it! KEYO typing checker program. Utility to give binary load files from Basic. Reviews of TopDOS, Homeword and Mr DO! Overview of FORTH as an alternative to Basic. Utility to fill in shapes in Graphics 8 and fast too! Profile on Lea Valley Atari Club. HAPPY TYPER gives automatic line numbers and programmable function keys. Utility for indexing Multiboot disks.

Issue 10
Includes all the facts about Digitised Pictures. Disk Jacket, a neat program for making your own disk covers. Opening Out, more about how files work. Reviews of the Great American Road Race, Kennedy Approach, Red Moon, Asylum and Wishbringer. Two excellent games PCB Paranoia and 3D-Maze. Introduction to the world of communications. Continuation of regulars Cracking the Code, Starting from Basics and What's MIDI?

Issue 11.
Includes RAM Talker for 400/800 machines. Reviews of Atariwriter Plus, Sidewinder, Koronis Rift, Electraglide, Mercenary, Fighter Pilot, Goonies and Alternate Reality. ST MIDI programs and ST hi-res Hat. Hexadecimal Code generator and some book reviews. Plus Starting from Basics and Cracking the Code.

ATARI ST

Power Without The Price!

FREE SOFTWARE

When you buy one of the new Atari ST computers from Silica Shop, you will receive a large and varied software package free of charge. This package covers several applications and comprises a total of nine titles. All STs now have TOS/GEM on ROM, so the total list of free software is as follows:

1) GEM – *DR Desktop environment with WIMP (in ROM)*
2) TOS– *Trained Operating System (in ROM)*
3) 1st WORD – *Word Processor by GST using the GEM environment and multiple windows*
4) BASIC - *Personal Basic by DR (with manual)*
5) LOGO – *Logo language by DR (with manual)*
6) DOODLE– *Simple paint /doodle drawing package (works on mono / colour systems)*
7) MEGAROIDS– *Asteroids type game by Mr Atmax*
8) NEOCHROME – *A powerful colour paint and graphics package fairly useable with colour systems)*
9) CP/M EMULATOR – *Allows the user to run on any ST system*

3rd PARTY SUPPORT

The power and potential of the ST range of computers is causing a flood of new software titles, peripherals and accessories from third party manufacturers. Titles range from word processing to spreadsheet, games, graphics and games to database management - all with those easy to use drop down menus and windows. With the list of companies producing ST software including dozens of top names you can expect some first class titles for the new ST range. The following includes a select on of the third party manufacturers who have developed or are working on products for the ST range:

ABACUS	EXTENDED S/W	MICRO-EX INC	SUBLOGIC SYS
ACADEMY	FIREBIRD	MICROPRO	SCANDINAVIAN
ACCOLADE	FIRST BYTE	MICRO PROSE	SIERRA ON LINE
ACTIONSOFT	FIRST FORM	MICROPUBLISHING	SIR SOFTWARE
ACTIVISION	FLIP 'N' WRITE	MICROSPHERE	SPECTRE
ADVENTURE INT	SILENTSOFT PRINT	MILES COMP	SOFTLAMS
AMTIC	GST SYSTEMS	MIRACLE	SOFTLOGIK
AMERICAN COMPUTER	HABA	MIRAGE	SOFTWARE CLUBS
ARTWORX	HAYDEN	MONOCHROME	SSI
AUDIN STATE	HBMB	MOUNTAIN SOFT	SOFTWARE PUNCH
ATI	HIGHSOFT	MUSIC	SOFTWORKS
AUTOCLAIMT	HYPERCOM	MUSTRON	SOFTWORKS
AZTEC	INFOGRAFIX	NEWTRONIC	SPINNAKER
BATTERIES INC	INSOFT	OCEAN	ST SYSTEMS
BAYDON	ISLAND LOGIC	OSS	STRATEGIC
BEEBUG	KNOWLEDGEWARE	OSBORNE/MCG	SUBLOGIC
BETTER WORKING	KOMA	OSE	SUNDATA SERVICES
BLUE CHIP	LABYRINTH	OTTER VALLEY	SUNSHINE BOOKS
BYS	LEARNER	OZI	SUPPLEMENTAL
CAMELBIN	LEVEL 1	PAPPAS MUSIC	SYSTEMATICS
CHAMELEON	LIGHTSPEED	PARAMAX	TALENT
CHRISTIAN	LLAMASOFT	PHASE ONE	
CHIPSOFT	LOGOTRON	PHILON	TDI ARRAN
COMPUTER	MARTRONIC CORP	PLANNER	TK COMPUTER PR
CRL / SILICA MUSIC	MAP COMPUTERS	PLANTER	TOP EXPRESS
DATABOARD	MARK of UNION	PROGRESSIVE	TRIANGLE
DATASOFT SYS	MARK WILLIAMS	PROMETHEUS	TYNESOFT
DATA SYSTEMS	MARTIN CONSULT	PSYCHOTY	UNISON
DELTA TRA	MASTERFILE	PSYGNOSIS	VIP
DELTA PLUS PRESS	MEGAMAX	PSYCHOLOGIE	WISSEN MICROCHIP
DIGITAL INTERFACE	MICRODEAL	RADIOSHACK SYS	WHITE KNIGHT
DUFOSE PUBNG	METACOMCO	RAINBIRD	WINDHAM CLASSICS
ELECTRONIC ARTS	MICROLIN	REGENT	WORD AND ICON
FERGON	MICRODEAL	RISING STAR	XLENT

520ST-M

NEW 512K 520ST-M KEYBOARD - The new 520ST-M keyboard costs only £349.96 (VAT=£399) and is yet another step forward in the specification for Atari computers. The keyboard, which includes both an RF modulator and cable, allows you to connect it to an ordinary domestic television set via a standard TV aerial lead and with 512K RAM as standard, a free package of disk organising and business software. The TOS operating system and GEM graphics package are now stored in the 192K ROM chip which can only be altered by installing a new chip. This means that the operating system will automatically load in when you switch on the computer. In addition to the key features you will also need to purchase either a 360Kbyte disk drive (RRP £130+VAT) or a 360Kbyte disk drive (RRP £174 +VAT). Either disk drive will provide you with fast, high resolution (roll over) and it will emulate a single drive. If you prefer not to use your own TV set you may purchase the SC1224 colour monitor (RRP £299+VAT) which is an ideal companion for the 520ST-M. The 520ST-M keyboard which includes built-in 512K RAM memory is set to remain at its present price of only £349.96 (+VAT=£399). This price will allow you with an alternative to the 104ST-F but at a lower price. Additionally it features the advantage of the 520ST-M's built in modulator.

£347

1040ST-F

For the businessman and the more serious home user, Atari have produced the 1040ST-F computer. This is a machine which can be interfaced to a business environment as a stand-alone system or can support a maximum of nine computers as a terminal. The new machine (1040ST-F) comes with a massive 1Mbyte of RAM memory as standard. In the marketplace as it is the first personal computer with 1Mbyte of memory as it normally was £800. You can now have the 1040ST-F as a much cheaper colour system. The present one colour monitor is £299 (VAT = £9.185). With the new system at only £499+VAT (£114.85) this new 1040ST-F not only represents twice as much memory as the 520ST-M, but it also has a built in floppy disk drive. Other enhancements are the real time clock built into the chip and a socket for a genlock interface. The new machine is fully compatible with all existing ST software and will only one memory used. The 1040ST-F is also supplied with a free mouse controller and the complete 1st Word word processing package. Also supplied are the GEM Desktop, Personal Basic, Logo, Doodle, Neochrome and Megaroids. Atari have also announced that they will be releasing a new range of high resolution monitors which will be available for use with the 1040ST-F. Full details of these monitors will be announced shortly. The 1040ST-F was manufactured solely with business users in mind and as such is supplied with a monthly lease rental of £29.90 RPI (ex VAT). We have stocks of the 1040ST-F as we go to press and can supply you with your requirements for demonstration or purchase. Call into your local Silica Shop for further details.

1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX
(1st Floor) 52 Tottenham Court Rd, London, W1
Selfridges (1st Floor), Oxford Street, London, W1A 1AB

£799

THE ATARI EXPLOSION!

By now you've read the specialist computer press, you will have realised that there is no other company which is giving as much editorial space at the moment than company is Atari Corporation. Atari have been making the headlines since the launch of their new 16/32 bit range of ST computers. Led by the powerful figure of Jack Tramiel and under the banner 'Power Without The Price', Atari like manufacturing new computers as opposed to price - with the power to challenge firmly established market leaders. With the introduction of IBM compatibility, a CP/M emulator, a powerful new working system and a communications package for the new 1 we could potentially see IBM disappear. It will be fairly before there is an occasion of the magnitude which will see Atari placed firmly beside such names as IBM and Olivetti in the personal computer marketplace. Read on to update details of what Atari are doing, and how they are pushing their 'Power Without The Price' specifications, and the needs of the user better.

FREE CP/M EMULATOR

This newly announced CP/M Emulation Package will enable software written under Digital Research's Z80 CP/M operating system to be run on the ST family of computers. There are several thousand applications written for CP/M in the UK alone, and several of the major CP/M (Private Investment) houses may convert their programs to 3½" disk formats for the ST range. The CP/M emulation package is supplied FREE OF CHARGE by Silica Shop with all ST computers.

IBM COMPATIBILITY

To make the ST available to those businesses who currently run IBM systems, and are looking for a low cost expansion method, Atari have announced a co processor unit for ST computers. This processor will open the ST range to all IBM or IBM compatible software applications. The unit which attaches to the computers via the DMA (Direct Memory Access) port, contains an Intel 8088 processor with 512K of RAM and will accept a 5¼" disk drive. In its ST mode the unit will also act as a second disk drive facility, the user an additional 88K of memory. The IBM emulator unit should be available by late September. If you would like to be informed when it is released, please complete and return the coupon below. We will send you further details as soon as we have them.

20MByte HARD DISK £739

The new Atari hard disk for the ST range has just been released. At ST dimensions it easily takes a back seat to the massive size of many of the new external mainframe hardware. The memory size of the disk is a massive 20 megabytes (unformatted) with a data transfer rate of 1.33 Mbytes per second. At approximately £700 + VAT it is very little more than the price of a single or twin floppy disk drive and is far more than offers massive storage at a fraction of the cost.

NEW ST SOFTWARE PACKAGES

There are now third party software packages which have been announced for the Atari ST range. These are also now include DR Mail + Diflex, Lattice C, as well as HAB Base + DBase 2 clones. In addition PC Intercommunications (VT 100 emulator which enables you to use any ST as a terminal to connect to a mainframe or mini computer. Other programs include a Lotus 1-2-3 clone (see paragraph below).

VIP PROFESSIONAL - LOTUS 1-2-3™ CLONE

VIP is perhaps one of the most impressive programs to have been released so far for the ST range. VIP Professional is an extremely easy to use, menu driven integrated software package which is virtually identical in operation and commands to Lotus 1-2-3™. The same spreadsheet, graphics and database management and extremely powerful graphics are all combined in a single integrated package. What is more, VIP Professional costs only a fraction of the price of Lotus 1-2-3™ yet you can also type in the same commands to do the same things. Probably the most surprising feature of VIP Professional is that is fully compatible with Lotus 1-2-3™ macros so that you can run many of the programs which have been written for Lotus 1-2-3™. VIP Professional costs only £199.95 (+VAT £234.25) whereas VIP Professional costs only £199.95 (+VAT £234.25) whereas VIP Professional costs only £199.95 (+VAT £234.25). That's less than half the price! If you would like further information on VIP Professional, please complete and return the coupon below.

SILICA SHOP

ATARI **WE ARE THE UK's NOT ATARI SPECIALISTS** ATARI

As Silica we have been successful and established as the specialist in the UK market. We can stock the complete range of Atari products and offer a first class back up service. You can be reassured that we are a long established Atari dealer and are fully supported. Our mail order clerks are fully trained and experienced to help you make the right decision. Our sales staff are also available to discuss your requirements and help you with any queries that you may have. We are always pleased to deal with your problems and supply your every need with our specialist service. We aim to keep stocks of all the complete Atari hardware and software range in our warehouse. We also stock a wide range of Atari associated books and periodicals. As Atari specialists we are able to keep you informed with details of the latest releases. Ahead of their release dates. We also have a wide range of dedicated magazines. We can provide a full service to all Atari owners and are now firmly established as the UK's NUMBER ONE Atari specialists. Here are just some of the things we can offer our customers:

★ **FREE POST & PACKING ON MAIL ORDERS**
★ **FREE NEXT DAY DELIVERY**
★ **INFORMATION MAILING SERVICE**
★ **TECHNICAL SUPPORT TEAM**
★ **HIGHLY COMPETITIVE PRICES**
★ **AFTER SALES SUPPORT SERVICE**
★ **REPAIR SERVICE ON ATARI PRODUCTS**

If you already own an Atari computer and would like to be registered on our mailing list as an Atari computer owner, or if you are a person interested in buying an Atari machine, let us know. We will be pleased to keep you up to date with new Atari developments free of charge. So return the coupon today and begin experiencing a spec sial Atari service the rest of the world is none

SILICA HOTLINE ☎ **01-309 1111**

SILICA SHOP LTD, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX
SEND FOR FREE ATARI ST LITERATURE

To: Silica Shop Ltd, Dept ATCOC 0886, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

PLEASE SEND ME FREE LITERATURE
ON THE NEW RANGE OF ATARI ST COMPUTERS

Mr/Mrs/Ms: Initials: Surname:

Address:

Postcode:

Do you already own a computer
If so, which one do you own?

ATARI ST

TIME BANDIT

ONE DAY ALL PROGRAMS [illegible] BE THIS GOOD

microdeal

BOX 6[illegible]
CORNWALL[illegible] 4YB.
TEL: (0726) 6802[illegible] MICROD G